# računari

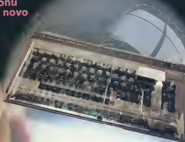
Specijalno izdanje časopisa „Galaksija"
Izdaje BIGZ – OOUR „Duga"

Izlazi jedanput mesečno
cena **300** dinara
februar 1986

**12**

*anđelko zgorelec*
**u londonu svašta novo**

*kad „spektrum" zataji*
**smrtonosna dijagnostika**



*dejan ristanović*
**kako rešiti „elitu"**

*pokice koje život znače*
**200 poukova za „komodor"**
**50 poukova za „spektrum"**

# 12

# računari

Izlazi jednom mesečno Izdaje BIGZ OOUR „Duga"
Cena 300 dinara / februar 1986. Specijalno izdanje časopisa „Galaksija"

**Izdaje**
Beogradski izdavačko-grafički zavod
OOUR Novinska delatnost „Duga"
11000 Beograd
Bulevar vojvode Mišića 17

**Telefoni**
650-161 (redakcija)
650-528 (prodaja)
651 793 (propaganda)

**Generalni direktor**
Dobrosav Petrović

**Direktor OOUR „Duga"**
Bratoljub Babić
**Glavni i odgovorni urednik**
Gavrilo Vučković
**Urednik izdanja**
Jova Regasek

**Tehnički urednik**
Mirko Popov

**Redakcija časopisa „Galaksija"**
Tanasije Gavranović, pomoćnik glavnog i odgovornog urednika
Esad Jakupović, zamenik glavnog i odgovornog urednika
Aleksandar Milinković, urednik
Jova Regasek, urednik
Zorka Simović, sekretar redakcije
Srđan Stojančev, novinar
Gavrilo Vučković, glavni i odgovorni urednik

**Stručna saradnja**
Dejan Ristanović
Dušan Slavić
Nevenka Spalević
Anđelko Zgorelec
Mihajlo Karapandžić
**Autori tekstova:**
Nada Aleksić
Branko Đaković
Donat Greber
Branko Hebrang
Đorđa Janković
Mihajlo Karapandžić
Vladimir Kostić
Vladimir Krstonošić
Saša D. Kovačević
Blažimir Miše
Dejan Muhamedagić
Ivan Nador
Radomir Nikolajev
Zoran Obradović
Miodrag Potkonjak
Dejan Ristanović
Jelena Rupnik
Dušan Slavić
Jovan Skuljan
Nevenka Spalević
Srđan Stakić
Zvonimir Vistrička

**Fotografije**
Vladimir Simević
**Ilustracije**
Miodrag Marković

**Prevodioci**
Esad Jakupović
Ksenija Pješčić-Lebedinski
**Izdavački savet „Galaksije"**
Dr Rudi Debijadi, prof. dr Branislav Dimitrijević (predsednik), Radovan Drašković, Tanasije Gavranović, Živorad Gligić, Esad Jakupović, Velizar Maslać, Nikola Pajić, Željka Perunović, prof. dr Momčilo Ristić, Vlada Ristić, dr inž. Milorad Teofilović, Vidojko Veličković, Velimir Vesović, Miloje Vuković
**Štampa**
Beogradsko izdavačko-grafički zavod
11000 Beograd, Bulevar vojvode Mišića 17
Žiro-račun kod SDK 60802-833-2463
Devizni račun kod Beobanke
60811-620-6-82701-999-01066
Za inostranstvo cena dvostruka (400 D, 2.50 US $, 6.50 DM, 45 Sch, 5.50 Sfrs, 20 Ffrs)
Na osnovu mišljenja Republičkog sekretarijata za kulturu broj 413-77/72-03 i „Službenog glasnika" broj 28/72, ovo izdanje oslobođeno je poreza na promet

## sadržaj

# Šta ima novo

**Autori: Neda Aleksić, Ivan Nador, Branko Đaković, Vladimir Kratonošić, Donat Greber, Dejan Ristanović**

## Komodor dizel

Firma Komodor izbacila je u Evropi još jedan nov model računara Commodora 128 D, duhovito nazvan 128 dizel. Reč je o modelu 128 koji je zajedno sa novim diskom 1571 smešten u odlično dizajniranu kutiju, dok je samo tastatura odvojena ali se može uvući u posebno ležište koje se nalazi na dnu same kutije, što olakšava prenos. Tastatura je ista kao kod PC-128, koja je nedavno proglašena za najbolju tastaturu kod kućnih računara a ceo uređaj prilično podseća na IBM-PC-a. Računar se može dobiti i sa ugrađenim hard diskom od 20 Mb, što ga zajedno sa odličnim CP/M-om i 80 znakova u redu čini vrlo interesantnim za poslovnu primenu. Cena uređaja je međutim možda najinteresantnija od svega — 1600 maraka (sa verovatnim padom u skoroj budućnosti) i predstavlja dosta primamljiv mamac. Razmislite dobro

## Japanska invazija

Invazija je počela ali na svu (ne)sreću samo na američko tržište personalnih računara. Nekoliko vodećih japanskih firmi je već izbacilo na tržište personalne računare koji su kompatibilni sa IBM Personal Computer, PC a pri tome su znatno jeftiniji. Seiko Epson Corp je startovao sa tri računara-Equity I, II, III koji odgovaraju PC-u PC/XT-u i PC AT-u a prodaju se za $1300, $1695 i $3495, čime je cena prepolovljena u odnosu na cene IBM-a. Sharp Electronics Corp lansirao je (malog) PC-7000 koji nije puno veći od portabl tranzistora, ali za cenu od $1795 opremljen je sa dva 5 1/4 inča flopi-diska 320K RAM-a displejom sa tečnim kristalom serijskim i paralelnim portom. Toshiba America Inc nudi računar težine 4kg sa 512K RAM-a displejom sa tečnim kristalom 3 1/2 inča diskom i paralelnim portom. Mi, bar sa sada, ne strepimo od invazije

**N. A.**

## Ploter za narod

Čuvena firma za proizvodnju igračaka Fischer poznata po svojim još čuvenijim kockicama uključila se u opštu računarsku groznicu od nedavno proizvodi interfeje koji omogućuje kontrolu modela pomoću računara. Interfejs je naravno napravljen prvo za komodor 64, a kada je interesovanje postalo veće prilagođen je i za ostale računare (MSX spektrum', BBC). Pored interfejsa u prodaju su pušteni i posebni računarski setovi sa delovima. Najinteresantniji komplet je ploter-skener. Nabavite komplet sastavite ga povežete ga sa interfejsom i računarom učitajte softver i — dobili ste odličan ploter formata A4! Uređaj radi kao i svaki drugi ,ozbiljan ploter — pomera olovku u svim pravcima diže je i spušta i zaista crta veoma precizno. Ako olovku zamenite s glavom za čitanje i promenite program dobićete odličan skener koji bi trebao da ,čita sliku koja se nalazi na papiru. Osim ovog može se nabaviti još nekoliko različitih kompleta za razne primene posebno u robotici. Možete napraviti grafičku tablu ili možda sebi partnera za šah. Prvi interfejs i pojedini kompleti već su stigli u našu zemlju (i u redakciju Računara ). Očekujte dakle detaljniji prikaz. Cena interfejsa je 250 nemačkih maraka dok se cena kompleta kreće od 300—600. To prilično znači da možete da nabavite solidan ploter za ispod 700 maraka. Sačekajte sledeći broj Računara pa onda donesite odluku

**V. Kratonošić**

## Nova cena

**Obaveštavamo čitaoce da smo zbog povećanih troškova proizvodnje pre svega cene papira i štamparskih usluga, bili primorani da cenu ,,Računara" povećamo sa 250 na 300 dinara. Čitajte nas i dalje — bićemo još bolji!**
**Redakcija**

## I digital u trci

Digital Equipment je rešio da još jednom odgovori na izazov IBM-a. Ovog puta je u pitanju personalni kompjuter kompatibilan sa IBM-ovim PC-jem. Predsednik DEC-a Kenet Olsen (Kenneth H Olsen) najavio je da će DEC-ov MS-DOS biti funkcionalniji i na žalost samim tim i nešto skuplji. Ako rešite da ga kupite ne tražite ga u velikim robnim kućama tamo se neće prodavati

**N A**

## Isto ali jeftinije

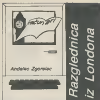
Tandy Corp upravo počinje prodaju novog modela — 3000. Računar je kompatibilan sa IBM-ovim Personal Computerom AT, ali će koštati oko 1000 dolara manje. Osnovna cena je 2599 dolara. Proizvođač iz Fort Vorta će, istovremeno utrošiti preko 80 miliona dolara da bi renovirao i učinio atraktivnijim prodavnice svog Radio Shack lanca i tako privukao kupce da se masovnije odluče za kupovinu Tandijevih personalaca iz kategorije koje se najbolje prodaje a to su prema analitičarima tržišta kompjutera, oni koji koštaju ispod 3000 dolara

**N. A.**

## Sastavite PC

Onima koji nemaju dovoljno novca a ne odustaju od IBM PC-a predlažemo da se odmah upišu na ubrzani kurs za sastavljanje hardvera. Za 1427 dolara može se kod ICS Inc (Puna adresa za kontakt i kupovinu ICS Inc 8601 Aero Drive San Diego Calif 92123 — tel 619-279-0084) nabaviti celokupni hardver za PC

**N A**






# u Londonu svasta novo

***Nesisvnim lansiranjem QL-a pre dve godine, januar mesec je, izgleda, postao tradicionalan datum za evropske premijere novih ili poboljšanih mikroračunara. U razmaku od samo nekoliko dana, ovog januara u Londonu su prikazani „amiga", navi „mekintoš" i četiri (na baš sasvim) nova računara firme Ejkorn (Acorn). Iako u znaku visoke tehnologije i izvanrednih-specifikacija, svaki od svih kompjutera na svoj način, s svi zajedno svojim visokim cenama produbljuju krizu koja razjeda industriju mikroračunara već više od godinu dana.***

Prošla godina je sa svet personalne informatike bile zaista crna. Mi smo vač opširno pisali o poteškoćama koje su pogađale čak i velikane na području mikororačunarstva. Najveća imena poput Ejkorna Sinklera, Atarije i Komodora ipak nisu propale Neke od ovih kompanije se čak manje-višę i uspešno bore sa krizom u industriji mikrokompjutera Prvi rezultati o predprazničnoj prodaji računara bili su čak i nešto bolji nego što se to očekivalo Organizacije za ispitivanje tržišta u Britaniji International Data Consultants smatra da su pesimističke prognoze davana još pre samo nekoliko meseci da će se u 1985 godini prodati samo blizu 900 hiljeda mikroračunara bile pogrešna i da ukupna prodaja prošle godina iznosi 1 1 milion malih računara (u rekordnoj 1984 godini prodato je čak 1 5 miliona mašine) Prognoze za 1986 godinu su još crnje — doći će do pada za 30% što praktično znači da će otići samo 800 hiljada kompjutera No traba sačekati baričetnu 1987 — eksperti IDC smatraju da će pojevom takozvanih „super" personalnih računara poput „amige" i atarijevih modela prodaja znatno porasti

## Ružna kao pas

Ova tmurna predviđanja međutim kao da se ne odnose na firmu Amstrad — njihovi računari idu i dalje kao elva. Ova firma zauzela je čak 25% britanskog tržišta i to na račun „staraca" Sinklera Ejkorna i Komodora Amstrad i dalje niže poslovna uspehe i dobije veliki publicitet nedavno je „boss" ove firma Alan Šuger proglašan kompjuterskom ličnošću godina a model PWC 8256, koji je i dosta teško i ovde nabaviti jer se odlično prodaje, uskoro će se naći na policama američkih robnih kuća — najveća američka trgovačka mreža „Sears-Roebuck" preuzima, u početku 10 hiljada ovih računara mesečno Ako prodaje dobro krene, „Sears-Roebuck" bi se dve godine trebalo da kupi od Amstrada čak pola miliona tih malih računara.

Od ostalih računara u toj glavnoj prodajnoj sezoni išao je jako dobro i „spektrum" koji i dalje zauzima vodeće mesto po broju prodatih mašine Spektrum uz put, izgleda jedino još u Jugoslaviji smatraju ozbiljnom mašinom i koriste ga za programiranje U Engleskoj je definitivno, prešao u kategoriju igračaka — ozbiljniji softver više niko niti otkupljuje niti prodaje

Komodorovi računari su i dalje popularni ali je firma ipak u velikoj finansijskoj krizi Početkom jenuara je zatvorena Komodorova fabrika u gradiću Korbi (Corby) u sradnjoj Engleskoj se gubitkom 300 radnih mesta a proizvodnje je prebačena u Nemačku (izgleda da će Komodor sada morati da vrati britanskoj vladi bespovratni zajam od nekoliko miliona funti koji je dobijen kao pomoć za otvorenje ove fabrika) Iz Amarika javljaju da je Komodor zatvorio i svoju fabriku za proizvodnju čipova u Silicijumskoj dolini i prebacio proizvodnju u jednu drugu fabriku na istočnoj obali Amerike Prema pisanju američkih poslovnih listova, dugovi su toliko veliki da ima malih izgleda da se ovaj pionir personalnog računarstva održi u sadašnjem obliku i da name „amige" Komodor bi izgleda, već potonuo

Kada smo već kod tog revolucionarnog računara da spomenemo da prvi korisnici nemaju baš najbolja mišljenje o njemu Prema enketi elektronskog biltena "Newsbytes" na "The Source" (elektronska datoteka u Americi, poput Prestele-Microneta u Britaniji) „amigi" je dodeljena titula „dog" (pas) zajedno sa užasnim PC Juniorom To je sve iznenadilo Traba međutim ponekad verovati i korisnicima „Amiga" je inače i službeno prikazana u Britaniji na sajmu "Which Computer" održanom sredinom jenuara u Birmingemu (Birmingham) i uskoro će biti puštena u prodaju iako osnovni softver još uvek nije obezbeđen

## Više čipova nego čips(ov)a

Lansiranje četiri nova Ejkornova računara iz serije „Master" koji su prikazani javnosti takođe sredinom jenuara u Londonu, na predstavlje na žalost ništa veselije događaj Novi računari su uglavnom zasnovani na poslovnom Ejkornovom ABC računaru prikazanom još pre više od godinu dana i u njima se zapravo krije stari dobri BBC B sa takođe poznatim proširenjima — dodatnim procesorima 6502 80186 ili 32016 Tehničke detalje prepuštamo ekspertu u radakciji za BBC računare Dajanu Ristanoviću koji za sledeći broj priprema prikaz sa obiljem komentara Za najnestrpljivija ljubitelje Ejkornovih mašine pripremili smo uporednu tabelu iz koje će im sve biti jasno

Na konferenciji se štampu bilo je skoro najmanje novinara kompjuterskih časopisa jer nisu bili namerno pozvani, a najviše iz listova opšteg profila Vaš dopisnik je nekako uspeo da provuče sebe i Vladimira Kostića, koji je tih dana bio u trgovinsko-programerskoj misiji u Londonu Predstavnik časopisa "Practical Electronics" i saradnik „Računara" Ejkornu nisu izgledali naročito opasno da bi im pravio smetnje

Konferenciju je otvorio generalni direktor Ejkorna Brajen Long (Brian Long) sa toliko samohvale i čestitki kao da ova kompanije pravi milione funti profita. Opšti je utisak da su sva četiri nove računara preskupa i da dolaze prekasno mada će se, ipak prodavati zbog velikog uticaje i popularnosti koju Ejkorn ima u školama i među dosadašnjim korisnicima BBC računara. Cene su gotovo neverovatno visoke — kao da Ejkorn na shvata da su druge kompanije

u poslednjih nekoliko godina drastično snizila cena uvođenjem složenijih i modernijih čipova što je smanjilo njihov ukupan broj Kako piše dopisnik mančesterskog The Guardiana u ovim novim računarima ima više čipova nego što ih izletnici pojedu na putu za popularno letovalište Blekpul (u engleskom chip' je naziv i za pržene krompiriće — pomfrit) Dok su ostale firme u poslednje četiri godine svojim računarima snizile cene i za 300% Ejkorn je u to vreme uspeo da poveća cenu za model BBC B od 320 na — za najjednostavniji Master računar — 490 funti Takva neposlovnost ili da budemo otvoreni drskost sigurno nije zabeležene u istoriji mikroračunarstva

Da je neka druge firme imale u svojim rukama model BBC B — koji je da se razumemo odličan računar i koji poseduje gotovo pola redakcije „Računara" uključujući i vašeg dopisnika — njegova cena bi sada bila oko 150 funti Sa takvom cenom BBC B bi imao šanse da izraste u najpopularniji racunar uopšte sa dve ili tri miliona korisnika a ima ih samo oko pola miliona Tada bi Ejkorn mogao da zaradi dosta novaca prodajom softvara i periferních jedinica i sigurno ne bi bio u finansijskim poteškoćama iz kojih ga je u dva navrata morao da vadi italijanski 'Olivetti'

Pitanje je da li će čak i škole kupovati ove nove bez sumnje dobre ali i preskupe računare bez obzira što za njihove daleko jeftinija konkurente ima manje obrazovnog softvara Dakle budućnost za Ejkorn je još uvek neizvesna i ukoliko Ejkorn ne uspe sa nekim drugim proizvodima, poput ROSC (reduced instruction set) čipova koji su odlično ocenjeni i koji su znatno snažniji i brži od konvencionalnih procesora a mogu da ih u potpunosti oponašaju (dakle, šansa za potpuno novi računar potpuno kompatibilan sa starim modelima) onda se Ejkornu loše piše! Uspeh na ovom polju mogao bi s druge strane Ejkornu ponovo da donese veliku prednost i nešto novaca U svakom slučaju firme bi trebalo da pokaže više poslovne sposobnosti a manje arogancije koja ju je i dovela na ovako niske grane Interesantno je šta važni londonski gradski finansijski krugovi misle o novim računarima Oni su pokazali svoje objavljeno oduševljenje time što su deonice Ejkorna odmah pale za 10% kada su cene novih računara A Ejkornove deonice su prošle godine zabeležile najveći pad od svih deonica na londonskoj berzi Neslavan rekord za jednu tako mladu industriju koja je mnogo obećavala!

## Novi „mek"

Za razliku od Ejkornovih računara najnoviji model mekintoša bio je jedna od najlošije čuvanih tajni Njegove specifikacije su bile poznate znatno pre konferencija za štampu održane u drugoj polovini januara u Londonu




Model makintoš plus ima nekoliko novitata koji ga čine veoma atraktivnim specijalni tasteri za brojke (numeric keypad) dva puta veća brzina udvostručena korisnička memorija na jedan megabajt duple (double sided) disk jedinica kapaciteta od 800K Uz još neke druge poboljšanja, u mek plus je ugrađen i specijalni interfejs nazvan Scusi (Small Computer System Interface) koji omogućuje da se na računar priključi čak sedam perifernih jedinica

Računar je već pušten u prodaju po dosta visokoj ceni od 2295 funti Originalni mek od 512K će se dalje i proizvoditi dok se onaj od 128K ukide

Dosadašnji vlasnici modela 512K će moći da nadograde svoju mašinu da radi poput meka plus" U Applu smatraju da za ove nadogradnje vlada veliki interes pa su zato obećali da će ovo proširanje prodavati po povoljnoj ceni" koja će kako mi saznajamo iznositi 300 funti Svi modeli makintoša za Evropu Aziju i Indiju proizvodiće se u novo modernizovanoj fabrici u gradiću Korku u Republici Irskoj






## Istinski bejzik

*Bejzik je nekada bio miljenče u obrazovnim institucijama i svaki početnik je počinjao upravo ovim jezikom za početnike kako mu i sam naziv kaže (BASIC je skraćenica od Beginner s All-purpose Symbolic Instruction Code)*

*A onda je pre više od decenije došao paskal sa svojim famoznim strukturalnim programiranjem i svako dalje pominjanje bejzika je postalo, najblaže rečeno nepristojno*

*Tomas Kurc i Džon Kemenj nekada profesori Dartmut koledža inače izumitelji bejzika kreću u ponovno osvajanje škola sa ovim potisnutim jezikom Verzije bejzika koju danas propagiraju ova dva profesora nazvana je istinski bejzik „Istinski bejzik može da uradi sve što može paskal i više od toga" kažu zagovornici tog novog bejzika*

*U doba kada smo razvijali bejzik na Dartmut koledžu postojale su komande koje su omogućavale strukturirano programiranje ali današnje verzije bejzika su sve to izbacile ostavivši svuda prisutnu GO TO komandu koju danas svi omalovažavaju Smanjenje broja komandi je samo jedan od primera kako komercijalizacija bejzika na mikroračunarima krši osnovne principe bejzika i kako se od originalnog bejzika sa koledža Dartmut stiglo do uličnog bejzika" kažu Kurc i Kemenj*

*Prema ovoj dvojici uglednih profesora sve je počelo u doba razvoja mikroračunara kada su resti koje uslovljene nedovoljnom memorijom računara, izazvale neophodna kresanje Pri pojavi moćnijih računara gde nije bilo ovih ograničenja, početne greške nisu ispravljene i to čine njih dvojica, danas, istinskim bejzikom*

*Da li će istinski bejzik potisnuti dobro ušančeni i danas najpopularniji Turbo Paskal videćemo u bliskoj budućnosti Pored toga što je veoma rasprostranjen Turbo Paskal se prodaje za 70 dolara, dok se za istinski bejzik traži 150 Većina današnjih računara se isporučuje se standardno ugrađenim „majkrosoftovim" bejzikom koji je možda slabiji od istinskog bejzika ali je praktično besplatan*

***I N***

## Kolor LCD

*Uskoro treba da se pojavi još jedna zanimljiva novost — kolor ekrani sa tečnim kristalima To najavljuju i Epson i Tošiba I kažu da bi cene trebale da budu duplo veće nego za običan LCD*

# Šta ima novo

## Greška, greškica

Misteriozni Komodorov PC je povučen tek što je predstavljen Očekuje se novo predstavljanje sada u januaru izgleda da su postojale dve suštinske greška — nešto nije bilo u redu u dovodu energija i računar nije primao dodatne pločice koje su spas za svaki PC Više sreće idući put!

## Koliko košta memorija

Veruje se da će uskoro cena memorijskih čipova porasti zato što se američki proizvođači potpuno povlače sa tog tržišta prepuštajući monopol Japancima Gde je monopol tu su i veće cene

## BBC je mrtav, živeo BBC

BBC B zaista ne postoji više ali je zato tu BBC+ Sjajan računar Jedino neki pakosnici i dalje tvrde da je cena prevelika

## Bitna Biblija

Zvuči neverovatno ali pojavile su se prve verzije softverskih Biblije Sad se zaista može reći da je nekim programerima računarstvo religije

## Uradi sam igru

Activision je proizvao programski paket Game Maker koji navodno, omogućava svakome ' da napiše super igru U paketu su Sprite Maker Scene Maker Sound Maker Music Maker i Editor Postoje verzija za epl II i komodor 64 Ovaj paket će biti jako popularan kod nas dok se ne sazne da Englezi ipak ne kupuju takve igre

## Prstom u računar

Epson je najavio LCD tablicu pomoću koje će se podaci unositi u računar direktno prstom Prava balkanska stvar!

## Ćaskanje sa ROM-om

Ubrzo posle puštanje u prodaju Taxi sistema (pisali smo o njemu u prošlom broju) koji je činio čuda za IBMov PC Epson je izbacio na tržište i svoj novi PC koji se zove pogodite kako Taxi PC Pogodite koji sistem koristi Lukavo

## Taxi za Epson

Keytronicsova nova dodatna tastatura koja je namenjena običnom PCju i XTju ima nekoliko novina Najznačajnija je to što se uz Keytronics 512 V kako se zove tastatura dobija i mali mikrofon pomoću koga se mogu direktno unositi naredbe Zgodno a?

## Sve su nam bliže

Vuman Computers Systems LTD je proizveo držite se čvrsto tekst procesor za ruski jezik Da li to znači da nam se polako primiču? Pitanje je kako bi to izgledao tekst procesor na jugoslovenskom jeziku

## Međuklase računara

Proizvođači pokušavaju da obrate pažnju na računarske međutržišta i kao rezultati takvih htenja pojavljuju se mnoge nove mašine kao što su Eprikotov XEN i Ejkornov Communicator

## Živi dalje

Evo još jednog dokaza da se kreativna ljude nama zastoja Anđelko Zgoralec pokretač prvog i još uvek najboljeg evropskog časopisa za kućne računare Practical Computer World i, naravno naš specijalni dopisnik iz Velike Britanije započeo je sa radom na novom časopisu koji se zove Practical Electronics Zasad toliko uskoro opširnije

## Deset maleckih

Časopis Practical Computing je napravio svoj spisak deset najboljih portabl računara To su Bondwell 2 Datavue 25 Data General One Epson PX-8 Husky Hunter Kaypro 2000 Olivetti M-10, Sharp PC 1500 Tandy 200 Toshiba T-1100 Cene vam nećemo pominjati jer su one dosta veće od samih računarčića

## Sa papira u memoriju

Da li ste nekada poželeli da unesete tekst u memoriju računara a da se ne gnjavite lupajući u tastaturu? Uplatite 400 funti firmi Oberon International (Hall Road Marylands Wood Estate Hemel Hempstead Hertfordshire HP2 7BH) i dobićete njihov novi Omni Reader Uređaj podseća na grafičku tablu u koju se umeće papir sa tekstom preko koga se zatim prelazi specijalnom optičkom olovkom Uz malo vežbe unosićete tekst brže od najbržeg daktilografa i to što je najlepše bez grešaka! Omni Reader se povezuje sa bilo kojim računarom uz pomoć RS232 interfejsa U cenu je uračunato i uputstvo za pisanje pratećg softvera ali ne i taj softver!

D R

## Dvostruko brže

Udvostručite brzinu vašeg BBC je za samo 20 funti poziva firma Solidisk Za ovu se naime sumu prodaje FourMeg tabla kojom se zamenjuje 6502 mikroprocesor osnovnog računara tako da kombinacija radi na cela 4 MHz Kompatibilnost sa softverom je stoprocentna ali će vam možda neke igre izgledati prebrzo U tabeli smo prikazali uporedne karakteristike raznih modela BBC-ja i drugih oglavnom šesnaestobitnih mašina pri izvršavanju standardnih 'benchmark testova BBC B dopunjen „FourMeg" tablom očito nadmašuje mnoge daleko skuplje kompjutere

Telefon Solidiska je (0702) 354674

### PCW Benchmark testovi

| | | |
|---|---|---|
| 1 | BBC B sa 32016 koprocesorom i BBC bejzikom | 6 88 |
| 2 | IBM PC AT | 7 11 |
| 3 | Olivetti M24 | 7 66 |
| 4 | BBC B sa FourMeg tablom | 9 79 |
| 5 | BBC B sa dodatnim 65C02 | 9 83 |
| 6 | BBC B sa dodatnim Z80 | 10 42 |
| 7 | Apple Mackintosh | 12 15 |
| 8 | BBC B ili BBC B+ sa bejzikom 2 | 14 55 |
| 9 | Apricot PC | 16 70 |
| 10 | IBM PC (standardan) | 17 60 |
| 11 | BBC B+ sa bejzikom 128 | 17 90 |
| 12 | Electron sa bejzikom 2 | 20 55 |

Load
dragi računari

## Dole matematika!

Mislim da su Računari dobar časopis i da koriste svakome ko ima računar Ima međutim i nekoliko slabosti koje mu smetaju da postane izvanredan

**Kvalitet štampe**, posebno kolor slika je **očajna**!! Da li ste u broju 10 pogledali slike igara u rubrici 1985 igre koje ste najviše voleli ? Da li ijedna sličica liči na nešto više od zamrljanog četvorougaonika?? Za časopis vašeg ranga to je nedopustivo Pogledajte konkurenciju (mada za mene to niste čitam i jedne i druge)! Moj mikro za isti novac daje isti broj strana sa superkvalitetnom štampom A i njih muči nestašica papira loša boja itd Mislim da se u ovom slučaju ne radi o objektivnim teškoćama nego o šlamperaju u štampariji Znam da vi niste za to krivi ali ipak dve-tri packe momcima u štampariji bi popravilo stvari Zbog čega štampate tako pametan i koristan časopis na WC papiru?

Matematički softver je izuzetno beskorisna rubrika koju je Dušan Slavič svaka čast njegovoj zaluđenosti brojkama ) namijenio ne znam koma Ja sam diplomirani inženjer matematike kompjuterom se ne bavim profesionalno nego iz razonode ali iako sve te stvari vrlo dobro poznam teško mi sledi tok misli Dušana Slaviča to jest saznati šta on uopće želi pokazati tim gomilama jednadžbi Šta misliia koliko prosečnih Yu-vlasnika mikrića uopće treba nakakvu tačnost na moguće kompliciranih funkcija od 10—12 decimala? Takve stvari su potrebne samo velikim stručnjacima kojima je programiranje POSAO a ne hobi a takvi stručnjaci ne trebaju Računare da bi te stvari naučili Još nešto — ne slažem se sa Dušanom da to što on zna moraju znati svi programeri — moraju samo oni koji rade na području matematike Pa kompjuter se ne koristi samo u matematici zaboga! Shvatite da neko može biti profesionalni vozač autobusa ali da ne mora zato znati na koji način se u tvornici akumulatora dodaje antimon u olovne ploče (Dušan bi takvog vozača verovatno nazvao diletantom kvaziprogramerom — pardon vozačem — itd)

Dakle dole sa rubrikom Matematički softver i na tom mjestu dajte što više recenzija knjiga tekstova opisa programa i uputa za njih (ne za igre nego za korisna programa) poboljšajte kvalitet štampe i časopis će postati SUPER!

Shvatite da čitaoci traže upotrebljive i korisne stvari a ne uopćene prikaze nečega što nikad neće koristiti

Imam i jedan predlog za najavljenu školu mašinca za C-64 Molim vas nemojte počeli sa opisima raznoraznih LDA STA JUMP naredbi To smo već uspeli napamet naučiti Ono što nama fali su uputstva za korišćenje asemblera i monitora Znamo šta znači koja naredba ali kako da ukucam LDA a da to radi? Takva pitanja muče stotine nas koji smo upoznali kernal i mašinac upoznali ali ne znamo tehniku korištenja asemblera i monitora

Zato očekujem da ćete uskoro objaviti barem popis naredbi ako ne već i kompletna uputstva za SUPER MONGY i Assembler/Editor koji su najčešći programi na kaseti a služe upotrebi mašinca na C-64 Bilo bi zgodno to stvaliti na jedan umetak To bi uz malo opširnije objašnjenje nekog provjerenog majstora (npr Zoran Živolić) bilo jedna od najkorisnijih stvari

Još jedna molba — više konkretnih tekstova a što manje teoretiziranje u stilu računari u školama obrazovanje i informatika i slično Dosta su o školstvu napisali i upropastili!!!) naši vrli ministri za školstvo i dosta stvari su zabrljali iako smo ih svi upozoravali da ne valja tako Tako će biti i sa računarima u školama Zato razvijajmo domaću softversku pamet kroz tekstove u vašem i sličnim časopisima i kroz knjige i pustimo da političari kroje sudbinu naše dece u školama jer im i tako ne možemo ništa a svi kad-tad odu — naravno na višu funkciju

Još mnogo godina korisnog druženja sa vama želim svim čitaocima (i vama)

**Marković Mladen**
J Gazija 16
41 000 Zagreb

***Dragi Mladene, ljudi kao što i sami znate koriste računare na razne načine pa i za — računanje U seriji „Matematički softver" prof Slavič se bavi poboljšanjima najelementarnijih matematičkih funkcija na kućnom računaru Redakcija oseća velika zadovoljstvo što u njenim redovima radi ekspert takvog kalibra Što se tiče kvaliteta štampe sumnjamo da se to može rešiti sa dva-tri packa***

## Mi smo ponosni

100 Javljam se na mali oglas u rubrici Razmena pod šifrom Virginsoft Pošto sam stidljive prirode neću nabrajati svoje karakteristike Samo ću reći da potpuno zadovoljavam zahteve u malom oglasu Predlažem sastanak radi detaljnog upoređivanja hardvera Sastanak je ispred Computer Shopa u utorak 25 februara Radi identifikacije predlažem da oboje u rukama nosimo po primerak Računara 11

200 GO TO 100

**Šifra Trideset i dva bita**

***Mi smo zaista ponosni što spajamo dva mlada procesora Budite zajedno sve dok vas nekompatibilnost ne rastavi***

## HIMEM ili HIMEN

Sa interesovanjem sam pročitao tekst Podmlađeni BBC u Računarima 11 i naišao na jedan zanimljiv detalj jedna od sistemskih promenljivih se u čitavom tekstu i u tabelama zove himen malo čudno zar ne? Razmišljao sam i nisam mogao da se setim kakva bi to skraćenica mogla da bude

**Čitalac iz Beograda**

***Sistemske promenljive BBC-ja koje određuju početak i kraj slobodne memorije se zovu HIMEM (od High MEMory) i LOMEM (Low MEMory). Ne želeći da se upuštamo u asocijacije BIGZ-ovih slovoslagača i korektorki srećni smo što Dejan nije pisao o BBC-jevom ponašanju u trenucima kada napažnjom programera podaci probiju gornju granicu slobodnog RAM-a***

## Podijum za saradnike

Primećujem u Računarima imena mnogih novih saradnika od kojih mi se trenutno najzanimljivijim čini Radomir Nikolajev iako to ne znači da ostali nisu dobri Naprotiv odlični su Zato mi je i pala na um jedna ideja Kako bi bilo da svakog novog saradnika predstavite manjom ako ne i sasvim malom biografijom a i slika bi bila interesantna Tako biste pružili priliku nama čitaocima da malo bolje upoznamo ljude čije članke tako rado čitamo Ja lično bih voleo da malo više saznam o Dejanu Ristanoviću

**Ivan Krtinić**
Sigel Zagreb

***Malo strpljenja biće i toga!***

## Ponovo Mirjana

Uzela sam ovaj najnoviji broj Računara i studiozno ga prelistala Nadam se da ova kritika neće biti i kritizerska

Prvo članak Šta ima novo je sažet i kompletan svaka pohvala za njega Nastavite samo tako

Drugo LOAD Dragi Računari — ima tu dosta interesantnih predloga (ako ne računate i moje) možda bi trebalo da se tim mladićima i devojkama i lično razgovarati Možda bi mogle i neke lepe ideje da se iskopa iz njih

Treće Računari u izlogu je korisno do onog dela koji govori o prenosu preko carine O uvozu tu nema ni govora

Četvrto Računari i obrazovanje Mislim da ste se tom temom otpočeli sa pogrešne strane Po mom mišljenju prvo je potrebna jedna kritika postojećeg stanja pa tek onda nešto više o budućim planovima

Za Peek & Poke Show sve same pohvale

Prošlog puta nisam imala konkretan dokaz ko su ti ljudi koji vole da nam sole pamet ali ne dam se da ste shvatili da se to odnosilo na Andriju Kolundžića

Pišite više o inostranim časopisima koji mogu kod nas da se nabave citirajte članak iz njih

Za mene je pun pogodak u metu članak Zorana Obradovića Kako mogu da stupim u kontakt sa Zoranom? Da se javim redakciji?

Vrlo malo pišete o knjigama a to bi bilo veoma korisno

**Mirjana Nikolić**
Bulevar Revolucije 255
11050 Beograd

***Miro razmišljamo da odvojimo jedan stubac u kojem bi izlazila samo tvoja pisma Imaš predloga komentara i kritike koliko i tri prosečna pismopisca Jednog od svih dana stupićemo u kontakt sa tobom***

## Svi Amstradovi ljudi

Javljam vam se povodom umetka Sve Amstradove rutine Prilog je zaista odličan i očekujem da ih njega izdate kao knjigu što ste već uradili za spektrum i komodor Ne mogu da čekam mesec dana da izađe narudžbenica

U Računarima 7 i 8 objavili ste dva odlična članka Zvonimira Vistričke o grafičkim mogućnostima Amstrada U poslednja dva broja ni traga ni glasa od Zvonimira Zašto

U Računarima 7 čitalac Dragan Grbić iznosi niz nedostataka o Amstradu Rekao bih samo nešto o pregrevanju monitora Ja imam zeleni monitor i kada sam ga držao u sobi gde je prosečna temperatura 25°C monitor se obavezno pregorevao posle 2—3 sata rada Sada ga držim u sobi sa normalnom temperaturom (oko 18°C) i samo mi se jednom pregrejao posle 8 sati rada

U Računarima 10 Dragan Miljković braneći svoj QL napada Amstrad On navodi da je Amstrad poljuljao QL-a samo u Beogradu da li je to tačno pokazuje činjenica da je Amstrad prodao od aprila 1984 preko milion računara dok o prodaji QL-a rečito govori činjenica da je Sinkler pred bankrotstvom Dragan i sam kaže da je pravo čudo što je Amstrad uspeo da izvuče iz Z80 ali je isto tako čudo što Sinkler nije uspeo da izvuče iz Motorole

**Goran Ilić**
19 septembra 1
34300 Aranđelovac

P S Da li će se Računari i dalje boriti protiv pirata pisanjem psihotestova i sličnih humorističkih tekstova ili će smisliti nešto novo Na jednoj strani ste protiv malih pirata a na drugoj zgrćete lovu na njihovim malim oglasima

## Nevinost i kako je steći

Čitajući tekst Podmlađeni BBC u Računarima 11 od Dejana Ristanovića otkrio sam da taj podmlađeni BBC ima nešto što je u vašem tekstu nazvano Himen-page Ja sam sve do sada živeo u uverenju da se to zove Himem-page ali pretpostavljam da sam pogrešio

Valjda su podmlađivači BBC-ja toliko dobro obavili posao da on sada ime i himen — znači to je potpuno nova i nevina mašina. Hvala na ovom intimnom uvidu u detalje podmlađivanja BBC-ja

*Ljubomir Dragutinović Zemun*

## Dole pirati

*Da li ima ikakve šanse da se sve ono što je opisano u članku Piratska je toga pregolema dogodi i kod nas! Zaista mi je muka kad pogledam male oglase u bilo kom računarskom časopisu i vidim one silne oglase pirata. Koliko znam strani časopisi ne objavljuju više takve male oglase otkad je donesen zakon protiv piratstva. Ako se kod nas donese takav zakon da li ćete vi prestati da objavljujete takve male oglase? To biste mogli da uradite i pre donošenja zakona kao svoj doprinos uvođenju reda*

*Sead Puškalić Mostar*

## Noge 1

*More se priznati da vaše naslovne strane u Računarima 11 zaista ne liče na stroge i tehnicizirane naslovne strane koje čovek može očekivati od jednog stručnog časopisa ali zar vam se ne čini da ste malo preterali. Uskoro će neki ljudi početi da vam kupuju časopis iako ne znaju ni što su to računari samo zbog naslovnih strana. Meni se noge na naslovnoj strani ne sviđaju ali moja žena ne deli to mišljenje. Imao sam problema dok sam je uverio da vašu naslovnu stranu podržavam samo teorijski iz profesionalnog razumevanja*

*D.Š. Beograd*

## Noge 2

*Umalo da ostanem bez Računara 11 i to samo zbog onih nogu na naslovnoj stranici! Znate li da sam jedva uspeo da ubedim moje roditelje da vaši Računari 11 nisu još samo neka vrsta Starta? Tek kada sam im pokazao listinge u sredini i tekst Dušana Slavića poverovali su mi da ste vi ozbiljan računarski list. Moj tata je čak zatražio da mu pozajmim taj broj da bi detaljnije pogledao članak Dušana Slavića. Sad vidite kakve probleme varnim čitaocima prave vaše super naslovne strane. Gogi moj drug iz razreda tvrdi da to i nisu neke naročite noge ali ja sam ga pokiopio argumentom da su to do sada najbolje noge na nekom računarskom časopisu*

*Miško iz 45 bloka*

## Noge 3

Najzad ste uspeli da napravite najružniju naslovnu stranu.

Zna se čak i vinovnik tog dela. To je izvesni Vladimir Simović koji potpisuje i ostale fotose objavljene u vašoj reviji (srećom ne baš sve samo one loše!)

Posmatrajući tehničku stranu fotografije za naslovnu stranu dolazim do zaključka da je ona snimljena Smenom 8 i to vrlo loše podešenom. Zašto?

Cela slika osim monitora je nejasna i mutna a boje su loše razgraničene. Jedino heljine opravdava svoju nejasnost. Najvažniji deo slika natpis RAČUNARI 11 je O.K. ostatak je očajan (pokušajte da pročitate natpise na tasterima!)

Inače i te računaru pridodate ženske noge su vrlo problematične. Pomalo su kvrgave i krive a tome je još samo doprinelo loše i nepravilno osvetljenje koje je nanelo odlučujući udarac celoj slici. Svetlo pada sa desne strane i odozgo što stvara čudne senke i pokazuje ono što bi baš trebalo da prikrije.

U ekranu monitora se jasno vidi dokle seže bela tkanina ili papir na kojem stoje Schneider i ženska. Tu je i autor kao i njegova oprema što nam jasno govori da je ceo hepening održan u nekom studiju.

Devojka je mogla da obuje sandale i neke druge boje. Tek da se razdvoji ko je ko. I zašto je odsečen levi kraj tastature?! Zar je Schneider toliko glomazan da ne može da stane u jednu foto-kompoziciju?

Dalje koncepcija i raspored objekata su jednaki kao na naslovnoj stranici RAČUNARA br 8. Jedina razlika je u tom što je ova slerija nastala tehnikom seckanja iz stranih časopisa a mlađa je autorski rad. Sve me to podseća na onaj ružni termin šrafciger — industrija.

Zar ne umete da napravite dobru naslovnu stranu? Stariji brojevi su bili pioniri u tom poslu pa i niste bili baš načisto sa tim šta staviti na naslovnu stranu ali sada imate već poveliko iskustvo pa nam i povećajte estetski nivo i naslovne stranice i sadržine.

U vezi s tim imate još jednog rušitelja u vašim redovima koji nam svaki put servira istu priču. To je vaš crtač Miša Marković koji se ubi crtajući gole ženske i tokenizovane tastature a le ZX 81 slova neopako i preveći non-stop portrete Andrije Kolundžića. Ni ja ne volim Andriju jer mu je samo cilj da uzme lovu ali ni grešni Miša ne sanja da Andriji samo pravi reklamu.

Zato bih vam predložio sledeće ostavite mogućnost da i čitaoci naprave neku ilustraciju i da to bude stalna praksa. Naravno vaša je stvar da li ćete te radove honorisati ili ne ili recimo da otvorite konkurs za najbolji crtež ili animaciju napravljenu kompjuterom. Siguran sam da bi bilo dosta posla oko ocenjivanja radova.

Pošto sam ja ilustrator voleo bih da vam uredim poneki crtež ili slično naravno bez nadoknade. Ne volim da uzimam lovu od onih koje cenim i poštujem!

Na ostali tekstualni sadržaj revije nemam nikakvih primedbi već vam savetujem da tako i nastavite.

P.S. Nadam se da gorepomenute krive i kvrgave noge ne pripadaju Jeleni Rupnik jer nju vrlo cenim.

P.P.S. Izvinite zbog grešaka i slike zasad još uvek nemam laket procesor i printer što ne znači da ih neću uzeti.

STOJANOVIĆ DARKO
Josipa Debeljaka 14a
11132 Beograd

*Priprema Branko Đaković*



# Pogled izbliza licem u lice sa „atarijem"

```
Desk  File  Run  Edit  Debug
                LIST                               OUTPUT
 30   next a
 10   for a = 0 to 255
 20   print chr$(a) ; " " ,
 30   next a
 10   for a = 0 to 255
 20   print
 30   next
                EDIT
 10 for a = 0 to 255
 20 print chr$(a) ; " " ;
 30 next a
Ok EDIT
Ok list
Ok run
Ok edit
```

Džek Tremiel zaista nije preterivao kada je tvrdio da sa super modelom njegove firme može da radi čak i čovek bez bilo kakvog kompjuterskog i matematičkog obrazovanja. S druge strane čovek sa minimumom kompjuterskog iskustva će se dobro oznojiti dok ne dovede računar u radno stanje!

## Mišem po bejziku . .

Najpre, kao što i pristoji, treba uključiti računar, disk i monitor (ako ste nekada radili sa nekom Pekardovom mašinom znate da bi se ta operacija mogla izvesti i pritiskom na samo jedan taster) pre čega se u drajv stavlja sistemska disketa. Računar će se tada ozbiljno zamisliti dok će se disk besomučno vrteti skoro čitav minut i GEM će biti učitan. Sada računar odlazi u hibernaciju i čeka da se vi iz nje probudite — treba zameniti sistemsku disketu drugom na koju je snimljen čitav bejzik interpreter. Novo će učitavanje potrajati sledećih pedesetak sekundi a onda će se ekran promeniti — u prvom redu videćemo meni sa prijatnim izborom Desk, File, Run, Edit i Debug. Za početak treba koristiti miša da se kurzor dovede do reči EDIT a onda pritisnuti taster i ispod zaglavlja će se pojaviti novi meni iz koga treba izabrati opciju Start Edit. Tada ste došli do tačke na kojoj su se vlasnici spektruma i drugih osmobitnih mašina nalazili u momentu kada povežu kompjuter sa napajanjem.

Možemo da budemo vrlo zahvalni programerima firme Digital Research što su dopustili da se u EDIT modu naredbe kucaju sa tastature — moglo se desiti i da se svako PRINT bira iz menija. Pa ipak miša nećemo smeti sasvim da zaboravimo — čim otkucamo program ili izvršimo neku izmenu u njemu, moraćemo da putujemo do natpisa „Exit Edit" a onda i do glavnog menija sa komandom „Run".

Dobro, reći ćete, miš ponekad može da bude koristan. Koliko ste samo puta kucali LIST da bi vam deo programa koji ste želeli da ispravljate „pobegao" sa ekrana? Miš će vam omogućiti da skrolujete program kroz Edit prozor na gore i na dole i da pročitate sve naredbe koje su vam potrebne. Pogrešno! U edit prozoru se ne može skrolovati tekst (ili bar mi, uz divno napisano uputstvo koje se kupcima isporučuje u fotokopiranom obliku nismo pogodili kako se to radi) programe ćete tako morati da editujete primenom ekranskog editora koji je vrlo sličan „gelekajinom" (bolji je utoliko što se po želji nalazite u Insert ili Overtype modu) a zatim ćete svaku ispravku morati da potvrdite pritiskom na RETURN. Ispravljene linije automatski putuju kako u programsku memoriju tako i na programski prozor.

Pošto ste nekako otkucali program sa File ga smeštate na disk a zatim sa Debug tražite greške u njemu (dibager je stvarno izvanredan i vredelo bi mu posvetiti poseban tekst). Kada mislite da ste na tragu neke od njih ponovo aktivirate opciju Edit i tako u beskrajnom nizu — miš će sve vreme biti potreban a takođe i poveliki prazan prostor pored kompjutera za njegovo kretanje. Ako

*Atari 520 ST nije više samo pusti san domaćih hakera: već i po pionirskim malim oglasima na stranicama kompjuterskih revija možemo zaključiti da su prvi primerci ovog kompjutera na razne načine srećno prešli jugoslovensku granicu. Nedavno je i autor ovoga teksta bio u prilici da se nekoliko dana zabavlja novom Atarijevom mašinom i napiše nekoliko programa za nju. Krivica za činjenicu da su ti programi bili kraći od pedesetak linija je isključivo Atarijeva — kada u računar učitate sve programe koje je potrebno ostaće vam slobodno samo pet kilobajta RAM-a, manje nego što imaju vlasnici neproširene „galaksije".*

vam je kao što programeru i prilići, sto prekriven hrpama papira moraćete da manjate navika ili avaj računar

## . . . a bejzik iz neolita

Ako ste mislili da mašina sa moćnim procesorom i ogromnim operativnim sistemom mora da bude opremljena moćnim i brzim bejzikom pogrešili ste. Atarijev bejzik je zapravo prilično stara Microsoft-ova tvorevina dopunjena sa par interesantnih naredbi za rad sa grafikom i tonom (o njima nekom drugom prilikom) i dve nove kontrolna struktura — DO WHILE (petlje sa izlazom na početku) i označeni GOTO (umesto GOTO 1023 možete na primer, da napišete GOTO ucitaj što složičete se čini program mnogo čitljivijim). Sa druge strane Atarijevom bejziku je oduzeta Microsoftova numerika — radi se samo 5—6 tačnih cifara, poput „galaksije" U uputstvu je doduše pomenuta mogućnost rada sa promenljivima u dvostrukoj tačnosti koje se deklarišu sa DEFDBL ili završavaju tarabom (#) ali se ta mogućnost verovatno izgubila u transportu iz Nemačke. Za uzvrat računar ima naredbu VAL koja u uputstvu nije spomenuta (drugo je pitanje što ta naredba na radi dobro) i funkciju VARPTR koje se sećamo još sa starog TRS 80.

Atarijev bejzik poput svih neolitskih varijanti ovoga jezika, omogućava jedino pozivanje potprograma naredbom GOSUB i definisanje procedura u jednom redu (DEFFN). Promenljive kojima nije dodeljena vrednost će kada ih štampamo ili računamo sa njima, imati vrednost nula, što je veliki potencijal za greške koje nije nimalo lako otkrivati. Osim toga, naredba RENUMBER se veoma čudno ponaša ako se koriste reference na nepostojeće linije, a ni komanda AUTO nije lišena „dečjih bolesti" koje se javljaju ako se zahteva iole veća razlika između linijskih brojeva susednih naredbi.

Rad sa periferijom je pregledno ali u mnogim stvarima traljavo rešen štampač ekran i disk imaju različite i međusobno nezavisne komandna struktura što znači da će se sa budućim proširenjima komunicirati isključivo naredbama POKE, koje se i na postojećem Atariju 520 ST koriste za mnoge stvari posebno za pozivanje potprograma iz GEM-a i prenošenje PARAMETARA koje će koristiti funkcije operativnog sistema. Mašinski potprogrami se pozivaju naredbom CALL.

Prijavljivenja grešaka je tekstualno i prilično zgodno iako se svodi na mali broj poruka. Većina grešaka razuituje porukom Something is wrong i ponavljenjem linije praćene strelicom koje ukazuje na grešku — treća sličnost između atarija i galaksija

## Nikad ne kucaj FRE()

Za kraj ovoga prikaza smo ostavili funkciju FRE() koja kao i na svakom Microsoftovom bejziku daje programeru obaveštenje o raspoloživoj memoriji. Pošto učitate operativni sistem GEM i bejzik pripremite čašu vode i otkucajte PRINT FRE(0). Atari 520 ST je kao što svi dobro znamo opremljen sa pola megabajta RAM-a, pa bi se dalo očekivati da za bejzik bude slobodno najmanje njih stotinak. PRINT FRE(0) će vas uveriti u gorku činjenicu da Tremielove mašina može da primi najviše bejzik program dug 5 kilobajta, manji od programa koji bi mogao da se smesti u memoriju neproširene „galaksije"! Ako se odreknete grafika i usluga GEM-a, program ćete moći da produžite za 32 K ali — zašto ste onda kupili atari?

Male memorija za bejzik se osim toga prilično neracionelno koristi — režijski prostor je petnaest bajta po liniji što znači da će PRINT Računari pojesti četrdesetak bajta. Na atariju 520 ST se dakle mogu pisati samo najkraći bejzik programi koji će se uz sve to izvršavati mnogo sporije nego na nekom amstradu i jedva nešto brže nego na komodoru ili spektrumu. Moguće je da će se u budućnosti pojaviti nove verzije bejzika i drugih programskih jezika ali bi one morale da budu komplikovanije i samim tim obimnije od postojećeg bejzika, atari 520 ST će dakle morati da se oprema memorijskim proširenjem da bi se na njemu moglo iole ozbiljnije raditi

Sva u svemu nova otkrića o Atarijevom „super kompjuteru" predstavljaju pravo razočarenja. Iako su uslovi nabavke sve povoljniji (cena u Nemačkoj je sa 3000 DM pala na 2200, dok se na božićnim rasprodajama kompjuter mogao naći i za 1900 DM) bejzik i slobodan RAM predstavljaju ogroman korak nazad u odnosu na standarde koje su postavila osmobitna mašine. Čak i prikazi u najumerenijim stranim kompjuterskim časopisima iznose slične utiske — Tramiel će morati mnogo toga da promeni ako želi da njegov računar postane interesantan za nezavisne softverske firme i uspe na tržištu

**Dejan Ristanović**

*9/licem u lice sa „atarijem"*

*Softverska trpeza*

# novinar u svojoj kući

***Ako već duže vremena sanjate o svom privatnom računarskom listu, pojavio se pravi program za vas. Programski paket Njuzrum (Newsroom – na engleskom „redakcija") vam omogućava da obrađujete novinarske članke, pravite ilustracije, „prelamate" strane i, naravno, štampate sve to. Treba samo naći čitaoce.***

Newsroom ** Newsroom

Newsroom

1

news novosti :

- Program za novinare -

NEWSROOM radi

Fotografije :

Celim programom se upravlja pomoću džojstika što rad čini prilično jednostavnim. Rad se obavlja preko opcija koje biramo iz menija a radni prostor su prozori. Već to govori da je uložen veliki napor da se imitira rad „meka". Zato je potrebno veoma malo uputstava da bi se program koristio. To ne znači da je on zaista nalik „mekovom" sistemu.

Prizor posle učitavanja špice je zaista prigodan: pred vama je slika redakcije nekog lista (vašeg lista) i sve moguće opcije rada.

Prva opcija se zove „Banner". On omogućuje izradu zaglavlja vašeg časopisa. Na ekranu dobijete beli prozor na crnoj podlozi u kome možete da pravite zaglavlje.

Sledeća opcija je na ekranu označena kao marka (slika) i služi za unošenje ilustracija u novine. Ilustracije u obliku sličica se nalaze na dve diskete koje dobijate uz program. Diskete sadrža nekoliko stotina sličica sasvim različitog karaktera (od ljudi do raznih čudovišta). Kada kursor (koji je u obliku ruke) dovedete do znaka sličice (marke) i pritisnete dugme na džojstiku, program će zatražiti da disketu Clip Art ubacite u drajv i da pritisnete dugme dok je kursor na opciji OK. Tada se na ekranu pojavljuje spisak različitih imena pod kojima su sličice grupisane. Ime najčešće i doslovno govori o sadržaju slike. Pod jednim imenom je grupisano od tri do desetak sličica. Kada izaberete šta vas interesuje, dovedete kursor do imena i pritisnete dugme. Disk počinje da se vrti i na ekranu se pojavljuje veliki prozor sa sličicama. Kada pronađete sličicu koja vam odgovara, treba dovesti kursor do nje i pritisnuti dugme. Program vas vraća u radni prozor gde sličicu možete premeštiti po volji. Ako ne uspete da nađete sličicu koja vam odgovara, na raspolaganju vam je opcija „Photo", koja je na ekranu predstavljena kao mala foto-laboratorija. Pomoću te opcije možete da slobodno crtate, prepravljate gotove sličice i još dosta toga. Kada završimo rad na sličici potrebno je samo izabrati opciju koja je označena malim foto-aparatom. Blic sevne i vi imate vašu sličicu koju možete snimiti na disk i koristiti kao i one koje ste dobili uz program. Ako ste talentovani za crtanje džojstikom, ovo je prava stvar za vas.

Ipak, vi ne želite samo da crtate — u novinama se i piše. Pisanje možete započeti pisanjem naslova i potpisa uz slike, što se čini pomoću opcije koja je nacrtana kao mala olovka. Na raspolaganju su vam tri vrste velikih i dve vrste malih slova. Pisanje članaka vam omogućuje opcija „Pannel". To je mali specijalizovani tekst procesor koji će vam pomoći da svoj tekst oblikujete u stupce i blokove. Dok ste u „Pannelu" ne možete crtati, ali možete učitati neku od sličica sa diskete. Pošto program ne dozvoljava snimanje na njegovom disku, moraćete uz njega formirati Data disk na kome ćete držati sve naslovne tekstove i slično.

Kada završite sa pisanjem teksta, crtanjem i pravljenjem zaglavlja, stranicu definitivno formiramo pomoću opcije „Layout". Po velikom prozoru koji predstavlja stranicu možete slobodno da razmeštate „Pannele" teksta, slika i naslova. Kada stranicu formirate, snimite je na disk. Posao u redakciji je gotov.

Ostaje samo štampanje. Ono se radi opcijom „Press". Ako ste pomislili da će možda biti problema zbog neusklađenosti sa printerom, prijatno ćete se iznenaditi — na ekranu ćete dobiti izbor od preko trideset raznih printera. Samo pritisnite dugme nad opcijom OK i iz printera će izaći prva strana vaših (i zaista vaših) novina. U tom trenutku ćete saznati kako je lep osećaj biti i novinar i urednik i tehnički urednik i slagač i štampar.

Ako ste posle ovakvog opisa rada pomislili da program prosto guta memoriju, niste u pravu. Program na zaista impresivan način smešta podatke u memoriju, tako da cela jedna strana teksta sa nekoliko fotografija i zaglavljem zauzima na disku, verovali ili ne, nešto manje od 4K. Impresivno, zar ne? Opcija koju nismo pomenuli je namenjena razmeni podataka. Predviđena je za rad sa modemom, odnosno povezivanje više ovakvih programa sa raznih strana.

Moglo bi se reći da je ovaj program uspešna minijatura sličnih programa namenjenih „meku". Ono što ga čini uspešnim je jednostavnost korišćenja i relativno niska cena. On ne predstavlja neku ozbiljnu mogućnost za privatno izdavaštvo, ali je izuzetno zanimljiv za eksperimentisanje i „interno" izdavaštvo. Izvanredan je za male lične formulare i fanzine. Nekom hakerskom klubu bi pametna upotreba ovog programa mogla da potpuno izmeni status. Prava mera ovog programa je izdavaštvo za ukućane i drage prijatelje, što precizno definiše njegovo mesto: na pola puta između igračaka i ozbiljnih uređivačkih programa. Obavezno ga isprobajte, možda ćete otkriti u sebi uredničku crtu. Vredi pokušati.

***Vladimir Krstonošić***

## Računari iz mog ugla

# Mi sve možemo

Vi i vaš računar ste slični: ne koristite svoje skrivene mogućnosti. Živite u harmoniji sa svojim računarom i nimalo vas ne zabrinjavaju filmovi na TV u kojima kompjuter-monstrum jede žive ljude. Za sada vaš računar ima više razloga da se plaši vas nego vi njega. Nespokoj se javlja kada zavirite u računar i ugledate mnoštvo čipova i kada se suočite sa masom nerazumljivih programa.

Kada bi vam neko na vreme pokazao svu gomilu knjiga koju morate da pročitate u toku svog školovanja, ne biste pristali ni da živite, a kamoli da idete u školu. Tako je i sa kompjuterima. Zgranuti ste kvalitetom i zatrpani kvantitetom. Obeshrabreni pojedinac pristaje na ono što mu se nudi. Polako sebe svodi na običan kondenzator ili otpornik, mizeran deo sa ograničenom funkcijom. Ideal svakog društva je da svako zna svoje mesto.

Ovakav defetistički stav pojedinca je u potpunom kontrastu sa zvaničnom politikom: MI SVE ZNAMO i MI SVE MOŽEMO. To je najlepši mogući primer dijalektičkog zakona prelaska kvantiteta u kvalitet.

Ne sme se sumnjati u to da mi sve znamo i da mi sve možemo. U vicevima smo uvek bili bolji od Amerikanaca i Rusa. Ove dve tvrdnje (MSZ–MSM) međusobno se prožimaju: istinitost jedne dokazuje se istinitošću druge i obrnuto.

Situacija u kojoj ste vi zaslepljeni time što MI sve znamo i sve možemo je vrlo zgodna za pojavu raznovrsnih oblika mimikrije računarskih autoriteta. (Obratite pažnju: MI — to smo svi mi, a naročito neki. Piše se velikim slovom iz poštovanja prema nama.)

Učili ste iz biologije šta je to mimikrija. To je kada, na primer, bezopasan gušter izigrava zmiju otrovnicu, kao kada biste spektruma opremili tastaturom IBM PC-a.

Na primer, uobraženi naučnik-profesor koji odbija uopšte da komunicira sa svoga visokog postolja, jer je sve to sa malim računarima ispod njegovog nivoa. Vi nikada nećete imati priliku da se tamo popnete, kucnete po njemu i ustanovite da je šupalj kao spomenik.

Jedna druga, vrlo česta varijanta su lažni autoriteti koji nešto znaju polovično. Polutani se služe svim mogućim pomoćnim sredstvima da bi ostvarili što ubedljiviji utisak. Oni ponavljaju stvari koje je neko već rekao i, neznatno ih menjajući, šalju dalje u opticaj. Kruženje priče obavlja se po zakonima kruženja materije u prirodi (učili ste o tome iz Poznavanja prirode i društva). Povremeno polutan ubacuje u opticaj nove sadržaje u koje će biti spreman da poveruje kada ih ponovo čuje od nekog drugog. Snaga tvrdnje ne zavisi od objektivne istinitosti, nego od toga koliko je puta ponovljana.

Ovo su ipak defanzivne taktike u kojima se može naslutiti slabost. Nešto kao pomoćni izlaz u slučaju opasnosti. Pravi rasni kompjuterski kontrarevolucionar će izabrati potpuno drugu taktiku: krenuće pompezno i agresivno kroz glavni ulaz. Njegova je namera da dobro unovči ono čime raspolaže u ogromnim količinama — svoje neznanje. Šta više, on mora biti apsolutno siguran da pojma nema. Uz to mora posedovati čelične živce i odlučan karakter. To ne može običan čovek nego samo Supermen.

Dok je polutan nespokojan pošto je nesiguran u svoje znanje, dotle je Supermen savršeno miran pošto je siguran u svoje neznanje. To je neravnopravni susret priučenog blefera i blefera velikog stila. Gledaće ga pravo u oči i na prvi nagoveštaj nesigurnosti zasuće ga bujicom stručnih izraza i velikih brojeva, što će potpuno blokirati mozak sirotog polutana, naviknut na preračunavanje na pijaci. Razni debeli direktori mogli bi biti masan zalogaj ako bi Supermen poprimio obličje prodavca računarske robe. Isto tako on voli da konzumira i lakšu hranu, naročito kada vrši misiju kompjuterskog prosvetitelja.

Sa polutanima je, dakle, lako. Izgleda da ozbiljne teškoće nailaze kada se namerí na nekog ko je takođe siguran u svoje (ne)znanje. Ni tu Supermen ne može da izgubi, jer se verovatno radi o drugom Supermenu. Supermen Supermenu nije vuk: oni će mirno razmenjati najnovije tehnološka dostignuća u bleferskoj struci, i razići se svako na svoju teritoriju.

Supermena ima malo, ali pravih mudriša ima još manje. Verovatnoća da će se sresti i da će mudriša uspeti da ga razbuca je dosta mala. Supermen zna da je za nedovoljno upućenog posmatrača istinito ono što se na ubedljiv način tvrdi. Prema tome on će izgledati sigurno i onda kada bude govorio najveće gluposti.

P.S. Drugovi zgranuti pojedinci, nemojte misliti da je za prepoznavanje kompjuterskih blefera potrebna neka velika škola. Šaljite svoje predloge za Supermena godine na adresu Peek & Poke Show.

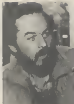
***Jelena Rupnik***

# Računari i obrazovanje

## Vaše dete govori bejzik, a vi?

***U poslednje vreme mnogo se priča o uvođenju računarstva i informatike u obrazovanje, ali je malo toga zaista učinjeno. Iz mnoštva lošijh imamo priliku da izdvojim i jedan izuzetan primer korišćenja računara u školi! Primer je lepši tim pre što se radi o nastavi likovnog vaspitanja dakle predmetu koji naizgled nema baš mnogo veze sa računarstvom***

Imali smo priliku da razgovaramo sa akademskim slikarom Mihajlom Marasanovim koji predaje likovnu kulturu u Osnovnoj školi Mate Balota iz Buja Na neki način on je unapredio nastavu likovnog vaspitanja uveo je računar tako da ucenici umesto četkice i boje koriste kompjuter kao sredstvo likovnog izražavanja Na taj



*Na sasvim drugi način Mihajlo Marasanov*

način učenici mogu bar donekle da upoznaju način rada sa računarom prve koraka u programiranju i si

Po mišljenju profesora Marasanova likovni rad izveden u tehnici računara može se svrstati u svako od klasičnih likovnih područja rada

Likovni rad može biti i animiran zvuk je također sastavni dio likovnog rada Vizuelne mogućnosti su neograničene kada računar povežemo sa videom

Mogućnosti su za sada skromne u školi trenutno postoji samo jedan računar i to spektrum 48 K sa INES tastaturom Nabavljen je prošle godine u maju i od tada se nastava likovne kulture izvodi isključivo pomoću kompjutera Uskoro će se naći sredstva za nabavku još nekoliko računara kao i jedne grafičke table Sledeći korak bio bi povezivanje videa sa računarom čime bi se upotpunila ova računarsko-likovna radionica

Sredstva za opremanja računarima ne bi trebalo da predstavljaju neki veći problem kaže profesor Marasanov Problem je u tome što samo u SR Hrvatskoj treba računarski opismeniti preko 25 hiljada nastavnike i oko 220 hiljada učenika što se po važnosti može uporediti sa posleratnim godinama kada je trebalo opismeniti preko 70% stanovništva.

U vezi sa tim profesor Marasanov i njegovi učenici su već nekoliko puta pozivali nastavnike iz škole Mate Balota" da nauče nešto o računarima i programiranju Štampali su pozivnice na kojima je pisalo Vaši učenici govore bejzik a VI? Mislim da sami možete predpostaviti kakav je bio odziv

Najgore što nam se može desiti je da ne živimo u svom vremenu Položaj nastavnike uključuje i više od toga, jer on mora da predvidi kako će živeti ta deca na koju on prenosi znanje Ne može se dozvoliti da učenik prvog razreda osnovne škole zna više od svog nastavnika To postavlja pitanje postojanje škola kao institucije što je sistemsko pitanja

Na osnovu svojih istraživanja, profesor Marasanov je ustanovio da je dete u uzrastu od pet godina već sposobno da samostalno komponuje bojama i grafičkim setom karaktera

Ako znamo da je učenik na nivou prvog razreda osnovne škole već rešio sve dileme oko oblika ili skupova, da je u drugom usvojio pojam promenljive onda je logično da u uzrastu trećeg razreda samostalno formira program Smešno je stoga da se tek pred kraj osnovne škola uvodi informatika kao neki poseban predmet kad se učenik već samostalno služi računarom Neoprostivo je koliko smo potcenili i omalovažili dete pripremivši mu ovakav program rada koji mu je bio primeren uzrastu u znatno ranijoj dobi Ali ako ne budemo radili po tom programu onako kako je to zamišljeno dobijamo u budućnosti ne sistemom školovane generacije već masu medioktiteta za koje ćemo mi verovati da su Pismeni a jedne strane i sa druge strane elitnu grupu opismenjenu mimo sistema škola koje će biti u stanju da drži generacijski korak sa svetom

Većina nastavnika s podozrenjem gleda na uvođenje informatika i računarstva u obrazovanje Ipak bilo bi lepo kada bismo u svakom broju Računara mogli da vam predstavimo po nekog profesora koji ne čeka novu reformu obrazovanje da bi u svojoj nastavi primenio kompjuter

***J. Rupnik***

# Domaći računari

## PC iz Podravine

***Domace tržište postalo je bogatije za još jedno osobno računalo, za „sperry PC 500" sklopljeno u radnoj jedinici Elektronika Drvne industrije Gaj iz Podravske Slatine Posljednjeg četvrtka prošle godine kamion za prijevoz namještaja umjesto uobičajene pošiljke prvi je puta iz tvorničkog kruga Gaja odvezao 64 PC-a Cijeli posao organiziran je u suradnji poznate tvrtke Sperry Univac zagrebačkog „Infosistema i Gaja Strani partner daje znanje u hardveru i sklopove Infosistem je servisor i kreator programske podrška za potrebe kupca dok „Elektronike zasad iz dobivenih dijelova sastavlja računala i opremu.***

Svako računalo marke sperry PC 500 sastoji se od sistemske jedinice s disk-jedinicom profesionalnom tastaturom kolor-monitorom a po želji kupaca isporučuje se sa štampačem ili ploterom

Pedesetak radnika Radne jednice -Elektronike koja je s radom počela prije godinu i pol se samo čenri je dana sklopila 64 računala iz prve serije PC-a 25 štampača i pet plotera

Sperry PC 500 koji će kasnije dobiti naljepnice Infosistema i vjerojatno gaj model A ima ROM od 128 K i RAM od 256 K s mogućnošću proširenje na 512 K Računalo je izvedeno profesionalno — tastatura je odvojena od centralne jedinice u čijem je sklopu i diskjedinice Monitor u boji upotpunjava taj kompjutorski set u mali poslovni sistem koji će na domaćem tržištu stajati pribižno 4 7 milijuna dinera Vani se prodaje se nekih 3 000 dolera



*Mali poslovni sistem Sperry PC 500 iz Podravske Slatine*

Ove godine planira se proširiti krug organizacije koje će raditi na tom programu Od domaćih proizvođača očekuje se osvajanje nekih pozicije hardvera — u početku mehanički dijelovi kasnije i elektronički Tako postoje izgledi da se ova računala oprema domaćim međuspojevima modemima štampačima Na području programa u posao će se uključiti i Đuro Đaković i neke druge organizacije koje već imaju formirane softverske grupe

Početna proizvodnje prva serija, unaprijed je prodana što daje realne izglede za nastavak Uskoro bi trebala biti sastavljena druge serija do stotinjak računala Kad se usvoji dio domaće proizvodnje pa iako smanje davaoje za uvozne komponente planira se godišnje proizvodnja od 5 000 komada PC-a iz Podravske Slatine Da li je to odviše smjelo pokazat će vrijeme!

Mnogi će se zapitati otkud elektronika u drvnoj industriji? Zašto šrafciger industrija "?

Prije dvije godine dva mlada inženjera iz razvoja službe Gaja Danko Potkonjak i Ilija Davidović predložili su da se ispita mogućnost razvoja na području elektronike Budući da je sirovinska osnova za drvnu idnustriju ograničena — Gaj je jedna od vodećih domaćih industrija namještaja građevinskih konstrukcija — odlučeno je da se podrži takva orijentacija Ubrzo je sagrađen pogon površine 1 400 četvornih metara Danas tu radi 50 radnika među kojima je osam inženjera i 15 kvalificiranih i visokokvalificiranih radnika. Oni koji prigovaraju još jednoj „šrafciger industriji nisu upoznati da se u većem dijelu svijeta računala i druge elektroničke naprave proizvode na bazi OEM tehnologije a da tek samo velesile u elektronici mogu proizvoditi čipove i druge suptilne komponente računala i opreme i dio komponenti što se sklapaju u Podravskoj Slatini nisu isključive proizvod Univaca već njihovih kooperanata

Naše šanse su zasad u proizvodnji mehaničkih dijelova, potom jednostavnijih elektroničkih sklopova i svakako programske podrške Tu bi trebali biti i najjači Primjera radi susjedi Mađari izvozeći softver i neke druge proizvode lani su uvezli automobile marke BMW

***B. Hebrang***

# Peek & poke show

## Računari ko Vas kvari!?

**Uvod**

*O računarima pišu ljudi a ne drugi računari ili pak neke vrste pitomih životinja i dok se u ozbiljnijim računarsko-specijalizovanim listovima i časopisima ne pojavljuju ljudi koji su sa tom vrstom sprave u najdaljim ako ne i vrlo zategnutim odnosima dotle je u takozvanoj revijalnoj porodičnoj megazinskoj i ostaloj štampi radiju a pogotovo na malim ekranima naših TV prijemnika situacija sledeća*

**Razrada**

*a) O računarima pišu ljudi koji su imali probleme i to velikih i sa računaljkama*

*b) Ti i takvi poznavaoci materije obično sarađuju i zarađuju u više glasila*

*c) Obučavali su se po skraćenim kursevima iz bejzika kobola tortrama kukičanja i važenja a u poslednje vreme vrlo je popularan tečaj Naučite kompi brzo klizanja i kuvanja i bićete najbolja partija u ulici, a možda i mnogo mnogo šire*

*d) Tako naoružani znanjem oni hrabro kreću u okršaje sa računarima i novim simpatizerima računarstva odvlačeći na krivi put mladu računarsku generaciju*

*e) Vrlo su glasni i sveprisutni baratajući sa oko osamnaest do dvadeset izraza koje koriste u svakoj pogodnoj i u dalu nepogodnih prilika istovremeno se kao orlušina obrušavaju na sve svoje protivnike ne dajući im da dođu do reči i dŕojstika*

*f) Vrlo su otporni na sve vrste poremećaje i promene klime a smatra se da će iščeznuti tek kada pokupe sve što se pokupiti moglo*

**Zaključak**

*Ovo je naravno sve bilo šala O računarima pišu i govore pametni obrazovani upućeni jednom rečju stručni ljudi A i meni se živi još koju godinicu*

## Neravnopravno takmičenje

*Sećate li se prošlog takmičenja malih (i maloletnih) šahista protiv kompjutera koje je na tako davno organizovao Đoka Vješti-ca to jest Studio B? Sećate li se onog klinca koji je matirao kompjutar u sedam poteza? E, pa takvo takmičenje se ponovo organizuje Predlažemo da klincima budu zavezane oči da igraju simultano protiv pet kompjutera i da im se za avo vreme borbi puste preglasna novokomponovana narodna muzika Zbog ravnopravnosti*

## PEEK & POKE SHOW

### Postavlja malo nagradno pitanje

# Čije noge se nalaze na naslovnoj strani „Računara 11"?

*Odgovore slobodno šaljite bez objašnjenja*

*Nagrade*

*Kao i uvek Peek & poke show je predvideo bogate nagrade za one srećnike koji pogode tačan odgovr*

*1 nagrada — Fotografija kompletne redakcije Računara snimljena pola časa pre nego što treba predati sve rukopise u štampu a pola ih tali*

*2 nagrada — Fotografija priređivača Peek & poke show u pozi sa naslovne strane računara 11*

*3 nagrada — Fotografija Jelena Rupnik ali nikome neće biti dodeljena*

*Pišite na adresu Peek & poke show Računari Galaksija BIGZ Bulevar vojvode Mišića 17 Beograd*

## Posle praznika

*Bilo je logično zapitati se da li će velika novogodišnje i božićna rasprodaja imati ikakvog uticaja na izbor računara kod nas Uticaj postoji i veoma je vidljiv — sva Komisionove radnje su prepune računara pogotovu starih VIC–20 i komodora 64 koji su za Novu godinu bili izuzetno jeftini Šta je sa cenama? Ništa iste su kao i pre Verovatno se i kod zvaničnih uvoznika može očekivati poveći kontingent računara (najviše komodora 64 ) kupljenih po nižim cenama samo što njih ne vredi ni pitati da li će cena kod nas biti nešto niže Oni to ne mogu iz principa da dopuste*

# Čip pobodi agency

## „Galaksija — Amerika"

*Agencija PPS izveštava da je firma Atari ponudila odgovarajućim institucijama u Jugoslaviji da im isporuči 50 000 komada računara atari 520 ST+ sa specijalnim popustom od 80% Kako se saznaje jugoslovenska strana se zahvalila na ponudi i odbila je iz principijelnih razloga Još uvek se ne zna da li je to učinjeno što bi postojale nejasnoće oko podele računara po republikama ili se možda radi o želji da se podrže izvanredni domaći računari galaksija lola orao MMS i omogući im se prodor na svetsko tržište*

## Vizionari iz „Računara"

*Kako se saznaje istaknuti društveno--politički radnici i nosioci odgovornih funkcija Saša Đakovič i Branko M Kovačavič pružili su punu podršku novoj reformi reformisanog školstva a posebno futurističkoj akciji Na svakih milion učenika jedan računar Ovaj novi podstrak usmeravanju presumereno usmeranog školstva će sigurno značiti puno za buduće generacije I nama sumnja da će se neka škola u budućnosti ponositi da ponese ime ovih vizionara*

## Visoka svest

*Evo još jednog primera našeg aktivnog učestvovanja u svetskim tehnološkim trendovima Radni ljudi Radne organizacije Jugospecijalitet' odlučili su da pošalju svog generalnog direktora sedam direktora i četrnaest šefova odeljenja na ubrzani kurs bejzika u Los Angeles u trajanju od šest meseci Time su naši radni ljudi još jednom pokazali da shvataju neophodnost uvođenja informatike i računarstva u sve grane privrede*

## Citat

*Anonimni TV citat „Ovo je „komodor 128" i on je dva-tri puta bolji od „komodora 64"*

## TV bez Modlija

*Televizija Beograd je obnovila svoje Nedeljno popodne Unutar tog nedeljnog popodneva obnovljena je i prošlogodišnja rubrika Program tvog kompjutera Na žalost rubrika je obnovljena tako što je izbačena jedina stvar koja je zaista vredela u njoj — a to je bio Zoran Modli Bez Zorana Modlija ta rubrika je zaista nalik ostalim emisijama RTV Beograd — bez mašte bez šarma*

***Pripremaju: Branko Đaković i Saša D. Kovačević***

# Računari u razgovoru:... sa dr Branislavom S. Bakićem

# zašto smo pali na niske grane

*• Druže Bakiću vi ste svedok korišćenja prvih računara u našoj zemlji. Pošto većina naših čitalaca u to vreme nije bila ni rođena, a o pionirskom vremenu primene računara kod nas skoro da nema sistematizovanog materijala, recite nam kada i kako je u našoj zemlji započela primena računara.*

— Početkom pedesetih godina, kada se u Zapadnoj Evropi javio širi pokret za proizvodnju računara, javio se takođe i kod nas. Tada je u SSSR-u počela proizvodnja računara Strela, Ural, Kiev, Minks, u Engleskoj ICT, u Francuskoj BULL, a kod nas se javio prvi računar CER koji je konstruisan u Institutu Mihajlo Pupin.

Međutim, dok je u SSSR-u, Francuskoj i Engleskoj nastavljeno sa industrijskom proizvodnjom računara, dotle se kod nas stalo na pojedinačnoj proizvodnji. Razlog za to je veoma jednostavan: naša industrija

**„Nivo primene računara u statistici bio je visok. U Evropi sigurno na samom vrhu"**

nije bila na tom nivou da bi mogla da proizvodi računare u većim serijama.

Akutne potrebe za računarima (popis stanovništva 1953. godine) dovele su do uvoza računara tada najpopularnijeg proizvođača IBM-a, što je za duži period odložilo domaću proizvodnju računara.

Mada zainteresovanost korisnika za primenu računara u to vreme nije pred industriju, a time ni pred nauku, postavljala izričite i uporne zahteve za postojanjem većeg broja računara, interesenti za primenom računara zadovoljavali su svoj interes uvozom, koji je bio daleko jednostavniji nego razvijanje domaće proizvodnje.

Potencijalni naučni kadrovi odlazili su u inostranstvo i tamo nalazili odlične uslove za naučni rad i život, pa se najčešće više i nisu vraćali u domovinu.

U tom periodu pedesetih i šezdesetih godina bum računarske tehnike bio je na Zapadu tako veliki da su ne samo naučni radnici nego i obični tehničari i programeri bili na visokoj ceni, što je privlačilo naše ne samo potencijalne naučne kadrove nego i sve druge stručne kadrove koji su bili u bližoj ili daljoj vezi sa računarima da odlaze na rad u inostranstvo. Tako je u našoj zemlji ostalo veoma malo kadrova koji bi mogli da se bave istraživačkim i inženjerskim radom u oblasti računarstva.

*• Ovog meseca februara 1986. godine navršava se 35 godina otkako su računari iz laboratorija gde su korišćeni samo u vojne i naučno-tehničke svrhe izašli u svet i krenuli u rešavanje jednostavnijih ali ne manje važnih poslovnih problema. Prvi komercijalni računar Ferranti Mark I, kome se ubrzo pridružuju UNIVAC i niz drugih serijski proizvedenih mašina projektovanih specijalno za poslovnu obradu, načinili su preokret u načinu poslovanja velikih preduzeća. Nakon 16 godina, u februaru 1967. godine pojavila se i prva domaća serija od 16 računara CER-200. Od tada je proteklo dovoljno vremena da bi moglo da se sagleda kako se kod nas razvijala poslovna primena računara i gde smo danas u poređenju sa razvijenim zemljama. Možete li nam dati neke kvantitativne pokazatelje koji bi ilustrovali uspešnost ove vrste primene računara?*

— Mada naučno-istraživački rad u oblasti računarstva posle prvih uspeha radova Dr Rajka Tomovića i računar CER nije doživeo procvat, u oblasti primene računara bilo je parcijalnih uspeha.

Primena računara u statistici počela je u Novoj Jugoslaviji vrlo rano. Prvi centar za

**„U osamdesete godine smo ušli sa primenom računara 16 puta manjom nego u SR Nemačkoj ili dva puta manjom nego u Čehoslovačkoj"**

obradu podataka počeo je sa radom još 1948. godine. Već 1953. godine svi Republički statistički zavodi i Savezni zavod za statistiku raspolagali su sa tada najmodernijim mašinama za obradu podataka. Nivo primene računara u statistici bio je vrlo visok. U Evropi sigurno na najvišem nivou.

Međutim, u drugim oblastima, naročito u privredi, primena je bila dugo vremena na najnižem mogućem stupnju. Tek 1965. godine počinje se sa primenom računara u industriji. Tada su samo tri industrijska preduzeća primenjivala računare u poslovima rukovođenja. Deset godina kasnije, 1965. godine, u našoj zemlji se primenjuje 45 elektronskih računara (približno 2,5 sistema na milion stanovnika). Još deset godina je bilo potrebno da se taj broj poveća na 1494 (1975. godine), uključujući mini računare. Taj broj predstavlja nivo od 25 računara na milion stanovnika, a tada razvijene zemlje imaju nivo: SAD 797, SR Nemačka 337, Francuska 249, Čehoslovačke 46. U osamdesete godine smo dakle ušli sa primenom računara 16 puta manjom nego u SR Nemačkoj ili dva puta manjom nego u Čehoslovačkoj.

Dok je u periodu između 60. i 80. godine glavni problem u široj primeni računara bio u nedostatku sredstava za uvoz računara (tako se svima činilo, jer nije bilo dovoljno sredstava pa se logično to uzimalo kao prvi i jedini razlog), dotle se u sadašnje vreme glavnim problemom smatra nedostatak kadrova potrebnih za širu primenu računara. Danas tzv. domaća proizvodnja nudi daleko veći broj računara nego što se oni mogu plasirati u praksi. Organizacije udruženog rada nemaju sluha za primenu računara, a tamo gde i ima interesa nema spremnih kadrova, na primer organizatora i projektanata za uspešnu primenu računara.

*• Vaša uža specijalnost je upravljanje poslovnim sistemima uz primenu računara, pa nam se čini da merodavno možete da kažete može li i kako primena računara pomoći oporavku naše privrede.*

— Savremena privreda zasnovana je na primeni računara. Ako naša privreda želi da ravnopravno učestvuje u svetskoj privredi, mora da na isti način koristi računare kao i ona. Ta istina prihvaćena je kod nas samo u jednom delu, i to u delu industrijske proizvodnje, pa se uvozi oprema koja je zasnovana na primeni računara. Međutim, u mnogo značajnijem delu, u upravljanju poslovanjem, svest o neophodnosti primene računara još nije prodrla u široke mase kadrova koji upravljaju privredom. Primena računara u čitavom procesu proizvodnje, a naročito u upravljanju, omogućava da se proizvodnja uzdigne na viši svetski nivo i time čitavo društvo izvede iz ekonomskih teškoća u kojima se sada nalazi.

Primena računara omogućava da se bolje izabiraju proizvodi koji će se proizvoditi, da se pravi više varijanti planova proizvodnje, da se utvrđuje stvarno optimalan režim poslovanja, da se neprekidno i bez ikakvog zakašnjenja prati izvršavanje optimalnog režima i plana proizvodnje i u slučaju potrebe reagira pravovremeno i na pravi način. Samo uz primenu računara je moguće proizvoditi na optimalan način, a to znači samo uz primenu računara može se postići željeni stepen korišćenja mašina, sirovina i radne snage.

*• Šta je po vašem mišljenju uzrokovalo da posle prvih projekata koji su obećavali blistavu budućnost naučnoistraživački rad u ovoj oblasti padne na današnji nivo?*

— Zanemarivanje razvoja naučnoistraživačkog rada u oblasti računara nastalo je kao posledica opšte klime nepridavanja važnosti domaćoj nauci nego oslona isključivo na uvoz. Uvoz kroz uvoznu proviziju

**„Primena računara u upravljanju omogućava da se proizvodnja uzdigne na viši nivo i time čitavo društvo izvede iz ekonomskih teškoća u kojima se sad nalazi"**

donosi sigurne materijalne koristi onima koji uvoze, a domaća nauka te koristi u devizama ne donosi, već na neki pijedestal uvaženih ličnosti dovodi neke ličnosti koje nisu iz kruga uvoznika.

Zanemarivanje savršenijih oblika domaće proizvodnje, pa čak i onih zasnovanih na uvozu, nastalo je kao posledica opšte klime rascepkanosti potencijalnih proizvođača. Ne samo da je svaka republika izabrala svoj tip računara koji namerava da proizvodi, nego su i pojedine regije nekih republika uzele svoje tipove. Za takav sistem proizvodnje nije bilo ni kod jednog od potenci-

***Razvoj i primenu računara u našoj zemlji prate mnogi paradoksi. Pre tridesetak godina imali smo izuzetno darovite konstruktore čiji su nas projekti svrstavali među vodeće u Evropi, a danas nemamo ni domaći računar niti racionalno koristimo mogućnosti uvezenih računara. Pokušavajući da utvrdimo zašto primena računara u našoj privredi i našem društvu nije išla uzlaznom linijom, potražili smo jednog od učesnika pionirskih dana korišćenja računara, Branislava S. Bakića, doktora ekonomskih nauka iz oblasti privredna kibernetika, dugogodišnjeg urednika časopisa Automatizacija poslovanja i direktora Centra za automatizaciju poslovanja. Branislav Bakić autor je više knjiga, priručnika, članaka i studija iz oblasti upravljanja poslovnim sistemima uz primenu računara i projektant brojnih projekata primene računara u organizacijama udruženog rada.***

jalnih proizvođača niti dovoljno kadrova a naročito ne dovoljno novčanih sredstava. Još 1962. godine Savezna privredna komora konstatovala je da 30 proizvođača želi vlastitu proizvodnju računara a to je svakako previše i za SAD ili Japan.

• *Od pojave prve serije domaćih računara cene ulaska u proizvodnju računara svake godine se dupliraju pa je danas po mišljenju nekih eksperata potrebno investirati više od 50 miliona dolara da bi se napravilo nešto što spada u informacijsku tehniku. Svima je jasno da ni jedan od pomenutih proizvođača ne može imati ni približnu količinu sredstava. Imamo li mi kao zemlja neku realnu šansu da se konkurentno uključimo u svetsku računarsku industriju?*

— Dosadašnji način snabdevanja korisnika računara bio je kroz uvoz. Zadnjih godina počinje se sa tzv. domaćom proizvodnjom. Poznato je da se proizvodnja računara može zasnovati uglavnom na četiri osnovna načina: preko vlastitog razvoja što je najbolji ali i najteži put, sklapanjem OEM ili kupovinom delova na svetskom tržištu delova pa sklapanjem u finalni proizvod po vlastitoj želji, proizvodnom kooperacijom pri čemu naši proizvođači proizvode neke delove a inostrani kooperanti druge delove koji se sklapaju u celini po projektu inostranog partnera, i licencnom proizvodnjom na bazi licence inostranog proizvođača. Naša domaća proizvodnja da-

***„Kroz uvoznu proviziju uvoz donosi sigurne materijalne koristi onima koji uvoze, a domaća nauka te koristi u devizama ne donosi"***

nas još nije prevazišla fazu čiste licencne proizvodnje, ali se javljaju veoma ozbiljni pokušaji da se proizvodi na bazi koprodukcije, sklapanja OEM a u oblasti automatizacije proizvodnje i na bazi sopstvenog razvoja. Može se reći da savremenu koncepciju proizvodnje računara ipak sačinjava shvatanje da će jedini pravi izlaz za domaću proizvodnju računara biti u oslanjanju na sopstvene snage u istraživanju i razvoju. Naravno, realizacija ove koncepcije zasniva se na udruživanju snaga svih potencijalnih proizvođača, jer ako do sada najavljenih više od 30 proizvođača u našoj zemlji ne shvati da samo kroz zajednički rad mogu da ostvare takvu koncepciju, uspeha neće biti. Vrlo je važno da se ovo što pre uvidi jer nas nepraćenje trendova u svetskoj naučno tehnološkoj revoluciji dovodi u zavisnost od razvijenih zemalja sa kojima trgujemo ili od njih pozajmljujemo kapital.

Jedini izlaz je da razvijamo domaću nauku u svim onim oblastima gde je to opravdano obzirom na perspektivu našeg razvoja. Informatika je svakako oblast u kojoj mi možemo da postignemo mnogo i to ne samo da se oslobodimo uvoza nego da postanemo i izvoznik. Mi imamo potencijalne kadrove, masu opšte školovanih kadrova koji se mogu specijalizovati za rad u ovoj oblasti. Razvoj informatike kod nas učinio bi mnogo za podizanje čitave privrede na viši nivo i za smanjenje naše zavisnosti od kolonijalne dominacije inostranih poverilaca i snabdevača.

• *Ali dok se mi usaglašavamo drugi ne sede skrštenih ruku. Tehnologija se sve brže razvija i jaz između nas i razvijenih svakim danom je sve veći. Kako ga premostiti?*

— Jedini put za smanjenje jaza u tehnološkom razvoju između razvijenih i nerazvijenih je podizanje svesti širokog kruga samoupravljača o posledicama tehnološkog zaostajanja. Ako oni koji odlučuju o investiranju kapitala misle da je jedini put njihove materijalne zainteresovanosti uvoz tuđih proizvoda, oni neće dozvoliti da se razvija domaća nauka i domaća proizvodnja ključne opreme. Da bi samoupravljači

***„Zanemarivanje savršenijih oblika domaće proizvodnje računara, pa čak i onih zasnovanih na uvozu, nastalo je kao posledica opšte klime rascepkanosti domaćih proizvođača"***

bili sposobni da oduzmu stvarno pravo odlučivanja o investiranju pojedinih vladajućim strukturama moraju postati svesni svoje lične zainteresovanosti za razvoj domaće nauke i tehnologije. To podizanje svesti samoupravljača je dug proces i nije zasnovan samo na verbalnom ubeđivanju nego i na praktičnom uveravanju na pozitivnim primerima iz prakse. No mislim da bi za čitaoce bilo mnogo interesantnije da umesto mog mišljenja o ovom pitanju čuju mišljenje onih koji se nalaze na mestima sa kojih se može dejstvovati na kreiranje politike.

• *Mišljenja onih koji su u poziciji da kreiraju politiku čitaoci i tako stalno slušaju, ali slabo se vide rezultati. Kao što nam se čini da bi svako od nas sastavio bolju fudbalsku reprezentaciju od državnog selektora kad ona zvanično gubi, tako verujemo da bismo mogli da uvedemo red i u proizvodnju i primenu računara. Stoga nam ipak recite koje biste mere vi preduzeli za smanjivanje jaza u tehnološkom razvoju između nas i razvijenih kao i u okviru jedinstvenog YU tržišta kada biste imali odgovarajuća ovlašćenja.*

— Široka primena računara zasniva se na opštem shvatanju čitave društvene zajednice da je primena računara korisna za zajednicu, nužna u sadašnjosti a naročito u budućnosti, da je usmerena u korist a ne na štetu čoveka, da ga oslobađa mukotrpnog rada i omogućava mu da mu rad postane zadovoljstvo, a ne da ga ostavi bez posla.

Za stvaranje opšte društvene svesti o korisnosti primene računara neophodna je angažovanost čitave društvene zajednice. S jedne strane političkih struktura, a s druge strane stručnih kadrova svih struka. Da bi se to postiglo moraju se primeniti određene mere. Pre svega treba organizovati naučnoistraživački rad u pravcu razrade koncepcije primene računara u pojedinim oblastima i osnovati škole koje bi osposobljavale kadrove potrebne za razradu te koncepcije i njenu realizaciju u praksi. U postojeće srednje, više i naročito visoke škole obavezno treba uvesti nastavu o primeni i korišćenju računara. Takođe bi trebalo omogućiti školovanje naših kadrova van zemlje u školama, institutima ili preduzećima koja primenjuju računare sa velikim uspehom. Naročito u periodu dok naše školske, naučne i privredne institucije nisu u stanju da pruže potrebnu pomoć kadrovima.

• *Može li se prevazići apsurd da se za korišćenje računara u mnogim našim srednjim školama, pa čak i na nekim fakultetima, obučava bez računara?*

Ovo se može ublažiti osnivanjem jednog ili nekoliko eksperimentalnih preduzeća koja bi poslužila s jedne strane kao demonstracioni centri za upoznavanje društva sa idejom primene računara, a s druge strane kao laboratorije srednjih i visokih škola. Isto tako treba osnivati i specijalizovane organizacije udruženog rada koje bi se bavile pružanjem pomoći organizacijama koje se spremaju ili uvode računare. Važno je razviti širok pokret za podizanje opšte kulture u oblasti primene računara, naročito u pravcu upoznavanja čitavog društva sa koristima koje od njihove primene može da ima. Temelji ovog pokreta moraju biti pored stručne štampe i popularnih napisa naročito obrazovne ustanove svih vrsta kao i televizija. Međutim sve ovo neće dati rezultate ako se ne donese dugoročni plan razvoja primene računara koji treba uklopiti u plan razvoja čitave privrede i društva kako bi razvojni put ove oblasti bio jasan za sagledavanje svima, a naročito kadrovima koji treba da ulože velike napore, najpre u svoje lično osposobljavanje a zatim u vrlo težak i odgovoran praktični rad. Oni to bez jasne perspektive fiksirane u najvažnijim društvenim aktima verovatno, bar kako to pokazuje sadašnja praksa, neće biti skloni da urade.

• *Recite nam još na kraju mislite li da sredstva informisanja mogu da doprinesu povećanju nivoa opšteg računarskog obrazovanja.*

— Sredstva informisanja su jedan od ključnih činilaca u stvaranju shvatanja o neophodnosti primene računara, pod uslovom da ovom problemu priđu odgovorno a ne senzacionalistički. Da objašnjavaju stvarnu korist od primene računara a ne da

***„Informatika je oblast u kojoj možemo da postignemo mnogo — ne samo da se oslobodimo uvoza nego i da postanemo izvoznik"***

prikazuju računare kao sredstvo uperено protiv čoveka ili kao nešto što može samo po sebi rešiti sve moguće probleme pridajući računarima moć koju oni nemaju. Nije teško da se uvidi društvena korist od primene računara i šteta od neprimenjivanja ako se čitava zajednica pravilno informiše o sadašnjoj eri i budućem razvoju primene računara, jer dosadašnji otpori primeni računara bili su posledica isključivo neobaveštenosti o koristi koje mogu imati od primene računara kako pojedinci tako i društvo u celini.

***Razgovor vodila Nevenka Spalević***

Istorija računara 35 godina

# IBM i sedam patuljaka

***Ovog meseca se navršava trideset i pet godina od pojave prvog komercijalnog računara. Ovo je priča o izlasku računara iz univerzitetskih laboratorija i njihovom pohodu na nove oblasti primene. Ko je IBM a ko su patuljci i zašto se baš tako zovu otkrićemo posle analize kako su se proizvođači snašli u poplavi zahteva za simbolom poslovnog uspeha pedesetih godina, a danas, van svake sumnje, neophodnim preduslovom za normalno poslovanje i uključivanje u svetsku podelu rada. Početak priče vezuje se za različita mesta — sve zavisi od toga ko pripoveda. Voleli bismo da možemo da kažemo da je pohod počeo sa brdovitog Balkana, ali držimo se činjenica i zato moramo reći — Počelo je na Ostrvu u našem komšiluku, a ne preko „velike bare".***

Mančesterski kompjuter MARK I uspešno je počeo sa radom u junu 1948 godine kao prvi računar na svetu sa unutrašnjim programom. Stvaranjem mogućnosti za korišćenje „Vilijamsove cevi" za memorijske uređaje, proizvedena je poboljšana verzija MARK-a I koja je korišćena za rad na optičkom dizajnu i za generisanje prostih brojeva. Britanska vlada, koja je nameravala da koristi računare u sopstvenim projektima za atomsku bombu, finansijski je podržala predlog svog šefa za nauku Sir Bena Lokspira (Ben Lockspiar) impresioniranog mogućnostima ovog računara da se komercijalna verzija MARK-a i proizvodi u lokalnoj mančesterskoj kompaniji. Tako je ova komercijalna verzija pod nazivom Ferranti MARK I bila na tržištu već u februaru 1951 godine, pet meseci pre UNIVAC-a, pa je možemo smatrati prvim komercijalnim računarom na svetu.

U Ferranti Mark I bile su ugrađene mnoge inovacije ali jedna od najznačajnijih bila je mogućnost modifikacije instrukcija za vreme izvršavanja programa. Sa ovom mašinom, koja je po mogućnostima bila ispred svog vremena, Ferranti je prvi ušao u kompjuterski biznis i prodao osam Markova u narednih nekoliko godina.

## Još jednom prvi

Džon Moučli (John Mauchly) i Presper Ekert junior (Presper Eckert), konstruktori ENIAC-a, prvog elektronskog računara na svetu, posle priznanja i para koje su dobili za svoj projekat, po starom američkom običaju osnivaju svoju kompaniju za proizvodnju kompjutera — Ekcert-Mauchly Computer Corporation. Međutim, vrlo brzo se pokazalo da bolje umeju da konstruišu nego da prodaju računare. Da bi mogli da realizuju svoju novu vizionarsku ideju — univerzalni automatski računar (UNIVersal Automatic Computer) koji bi mogao da se koristi i u biznisu, morali su da se nađu pod patronatom neke finansijski moćnije kompanije.

Na žalost, u to vreme većina ljudi, pa čak i onih koji su se projektovanjem računara profesionalno bavili, nije uviđala svrhu upotrebe kompjutera van univerzitetskih laboratorija, pa ni projekt UNIVAC-a nisu ozbiljno shvatali. Između ostalih, nije pristao da otkupi njihovu firmu čak ni IBM, najmoćniji proizvođač kancelarijske opreme. Srećom, Remington Rand, stari rival IBM-a, imao je više sluha za inovacije koje su nudili Ekart i Moučli i u februaru 1950. godine uzeo ih je u svoje okrilje. Koliko je to bio dobar potez pokazao je već popis iz marta 1951 godine za čiju je obradu korišćen prvi UNIVAC.

Osnivač IBM-a, Tomas Votson

### *Mislio je kratko*

*Kada je Vilijems (F. C. Williams), tvorac „vilijamsove cevi" (memorijskog uređaja na bazi katodne cevi) i profesor mančesterskog univerziteta na kome je načinjen niz značajnih inovacija u računarstvu, bio u upravi IBM-a prilikom svoje posete Njujorku, upitali su ga kako to da mančesterski tim tako produktivno radi na novim tipovima računara, tako se u njega ulaže znatno manje sredstva nego što se to čini u Americi. Budući da je i veliki IBM koristio mnoge od mančesterskih patenata, Vilijems je uz osmeh, pokazujući na jedan od mnogih omiljenih transparenata starog vlasnika kompanije Votsona na kome je pisalo „MISLI", rekao „Nismo se suviše zadržavali na razmišljanju."*

UNIVAC je imao nekoliko neobičnih svojstava za mašine tog vremena. Koristio je poseban program za prevođenje programa u mašinski jezik (compiler), a sami programi čuvani su kao i podaci u unutrašnjoj memoriji. To je bila prva mašina sposobna da obrađuje kako čisto numerička, tako i alfabetske podatke. Osim toga, ako izuzmemo britanski Mark I koji je imao znatno manje tržište, to je prva mašina sačinjena ne po specijalnoj porudžbini već za slobodnu prodaju.

Ipak, malo je kompanija koje su poput Remington Randa ikada izbacile tako izvanredan proizvod, a prodavale ga tako loše i nesposobno. Da nije tako bilo, možda danas ne bismo mogli reći da je priča o komercijalnim računarima, u stvari, priča o proizvođaču koji već više od 30 godina podmiruje preko 60% svetskog tržišta — priča o IBM-u.

## Saga o Votsonima

Firma koju je osnovao 1924. godine i briljantno vodio Tomas Votson senior (Thomas J. Watson 1874—1956) brzo je zauzela vodeće mesto među proizvođačima kancelarijskih mašina. Već 1939. godine imala je veći profit od svojih glavnih konkurenata Remington Rand-a, NCR-a (National Cash Register) i Boruza (Burroughs), a posle II svetskog rata i njen ukupni obrt kapitala bio je najveći. Tada već ostareli Votson senior nije bio naročito zainteresovan za ulazak u nove rizične poslove sa elektronikom i kompjuterima, ali svestan značaja reklame ulagao je sredstva i u finansiranje računarskih projekata. Tako je 1939. godine preuzeo finansiranje konstrukcije Automatskog sekvencijalno kontrolisanog kalkulatora (Automatic Sequence-Controlled Calculator) poznatijeg pod imenom Mark I Hauarda Ajkena (Howard Aiken). (Ovaj harvardski projekat nema veze sa desetak godina mlađim istoimenim engleskim računarom.)

Rad na harvardskom Mark-u I trajao je oko pet godina i avgusta 1944. računar je zvanično predstavljen. Želeći da zadrži za sebe najveći deo priznanja, Ajken je na toj ceremoniji jedva pomenuo IBM, što je strašno razljutilo Votsona. Njegov stariji sin i naslednik kompanije Tom Votson je kasnije izjavio: „Da su Ajken i moj otac tada imali revolvere, danas bi obojica bili mrtvi." No,

i pored ovog nesporazuma investicija od 500 000 dolara koliko je koštao Mark i i činjenica da je ovim računarom koji je imao sve osnovne blokove „analitička mašina" — uređaje za ulaz i izlaz memoriju i upravljački i aritmetički organ — ostvarena mašta Čarlsa Bebidža jako je podigla ugled IBM-a u akademskim krugovima Mada je Mark i predstavljao veoma skupu vežbu u zastareloj tehnologiji već prevaziđenoj izumom ENIAC-a, ipak je iskustvo stečeno na njegovom projektu i realizaciji značajno koristilo IBM-u u proizvodnji novih tipova računara

U skladu sa tipično konzervativnim pristupom umesto da ulaže u projekte elektronskih računara IBM 1945 godine ulaže sredstva u projekat Volas Ekerta (Wallace Eckert, šef kompanijina naučna laboratorije nema veze sa Presperom Ekertom) Selektivnog sekvencijalnog elektronskog kalkulatora — SSEC SSEC je bio tehnološki hibrid koji se sastojao od 12 500 cevi i 21 400 elektromehaničkih releja Cevi su korišćene za vršenje aritmetičkih operacije i beleženje male količina podataka i instrukcija (osam dvadesetocifrenih dekadnih brojeva) Releji su služili kao sporija ali veća unutrašnja memorija za 150 brojeva. Još 20 000 brojeva moglo je biti upamćeno

## Računar iz kuhinje

*Put do relejnog računara vodio je preko kuhinjskog stola — Džordž Štibic (George Stibitz) matematičar zaposlen u Belovim laboratorijama primetio je sličnost između brojenja pluseva i sabiranja. Radeći kod kuće na kuhinjskom stolu i koristeći nekoliko starih krosbar prekidača i elektromehaničkih releja načinio je prva relejna kompjuterska kola Štibic je tada potražio pomoć iskusnog inženjera Semjuela Vilijamsa (Samuel Williams) koji je već 25 godina radio na prekidačkim kolima i uskoro su ova dva naučnika stvorila relejni računar kome su dali naziv Comlex Number Calculator CNC počeo je da se koristi 8. januara 1940. godine*

na 66 koturova bušene papirne trake Instrukcije i podaci prebacivani su sa trake na releje, a zatim na cevi gde su se instrukcije izvršavale Mada je ovako izvršavanje instrukcije dosta trajelo relativno velika memorija davala je ovom računaru veće mogućnosti od ENIAC-a. Ovaj projekat nije imao skoro nikav uticaj na dalji razvoj kompjutera, ali ga pominjemo jer su neki eksperti razume se IBM-ovi proglasili ovaj računar završen u januaru 1948 godine prvim računarom na svetu sa unutrašnjim programom Ovo je samo delimično tačno istina je da su njegove instrukcije i podaci bili kodirani brojevima i pohranjeni u unutrašnjoj memoriji, ali računar nije bio potpuno automatski Koristio je nekoliko tabli sa utičnicama na kojima se mali ali značajan deo programa zadavo odgovarajućim povezivanjem žica, stoga mančesterski Mark završen nekoliko meseci kasnije ipak smatramo prvim računarom sa unutrašnjim programom

SSEC je bio instaliran u izlogu IBM-ove zgrade na Menhetnu i izazivao je veliku pažnju prolaznika. Ljudi su se okupljali pred izlogom i fascinirano posmatrali treptanja sijalica na kontrolnom panelu okretanja koturova trake i štampača koji su izbacivali podatke fenomenalnom brzinom od 30 000 cifara u minuti IBM-u nije bila potrebna bolje reklama.

Tvorci legendarnog UNIVAC a Presper Ekert i Džon Močli

Neobičan početak Prvi komercijalni računari u Engleskoj nastali su u ovoj prodavnici čaja

## Odbrambeni kalkulator

Zahvaljujući ovim projektima i nizu drugih manjih, Vladine agencije uputile su IBM-u zahtev da za njihove potrebe načini elektronski računar koji je u projektu imao naziv „odbrambeni kalkulator" (Defense Calculator) Međutim IBM nije hteo da Vladi bez patentnih prava daje projekat, pa su njegovi stručnjaci uradili finansijsku konstrukciju po kojoj bi se projekat i proizvodni program finansirali sopstvenim parama. Po njihovoj proceni, bilo je neophodno uložiti tri miliona dolara, a to je i za prebogati IBM bio veliki zalogaj U firmi su postojala dva struje jedna na čelu sa mladim Votsonom koji se kao vojni pilot upoznao sa mogućnostima radara i drugih elektronskih naprava i pripadao generaciji

## Prvi domaći računar

*Prvu domaću seriju računara CER-200 završio je bez pomoći industrije početkom februara 1967 godine institut Mihajlo Pupin' Još pre no što je bila završena nulta serije primljeni su zahtevi za izradu novih 50 mašina zahvaljujući ceni od 22 miliona starih dinara +4 500$ u vreme kada su uvozne mašine skromnijih mogućnosti koštale oko 28 000 dolara (oko 35 miliona starih dinara)*

projektanate ENIAC-a i bio „za" i druga konzervativna koja je smatrala da je bolje izvršiti ulaganja u već proverenu tehnologiju

Odsek za planiranje oštro se suprotstavio projaktu sa obrazloženjem da kroz pedesete godine ni jedan kompjuter neće biti iznajmljan po ceni većoj od 1 000$ a da bi se opravdala ulaganje cena iznajmljivanja bi morala biti bar 8 000$ Tom Votson junior se nije predavao Za svaki slučaj njegovi savetnici prikupili su 30 garantnih pisama budućih korisnika računara i u martu 1951 godine već je postojao projakat na papiru Videlo se da cene iznajmljivanja mora biti izuzetno velika čak 15 000$ ali kako je većina mušterija ostala pri svojim garantnim pismima odbrambani kalkulator' počeo je da se proizvodi 1952 godina Pre no što je njegova proizvodnja počela u IBM-u je bilo zaposlano samo nakoliko elektroinženjera a za naredna dva godina zaposleno ih je skoro 500 U tom prvom talsau zapošljavanje visokostručna radna snage hardvamši su bili na visokoi ceni

Prvi IBM 701 kako je odbrambeni kalkulator' iz projakta nazvan isporučen je Los Alamosu u martu 1953 To je bila binarna mašina u fon Nojmenovom stilu sa 4000 cevi read/write elektrostatičkom memorijom od 4 K 36-bitnih reči i mogućnošću da pomnoži dve broje za 456 mikrosec Računar je bio pun inovacije uključujući i tako trivijalne detalje kao što je pakovanje — bio je izrađen u delovima koji su pakovani u kutije od kojih je svaka mogla da stana u standardni lift i prođe kroz obične vrata IBM je konačno na velika vrata ušao u kompjuterski biznis Ni jednog trenutka njegovo tržište nije padelo ispod 50% u SAD i 60% svetskog tržišta Pod vođstvom Tomasa Votsona mlađeg koji je imao nepogrešiv instinkt za nove tehnologije i trgovačku umešnost svoga oca IBM je sve ove godina uvodio prava mašine u pravo vreme Tako je već 1955 proizveo računar u kome je 1250 elektronskih cevi zamenjano sa 2200 tranzistora čime je potrošnje energije smanjene za 95% a danas je u njegovim laboratorijama razvijeno nekoliko kompjuterskih elemenata koji koriste nova tehnologija na primar tehnologiju baziranu na Josephson efaktu Danas jedino Japan i Velika Britanija imaju nazavisne kompjuterske kompanija koje mogu da se takmiče sa IBM-om na sopstvenim tržištima Nećemo se zadržavati na japanskom tehnološkom čudu čije bi analize zahtevale mnogo više prostora, već ćemo se vratiti tridesetak godina unazad u Englesku da vidimo šta su to Englezi preduzeli što je omogućilo da sva vrama ostanu u vrhu kompjuterskog biznisa

## Lajonsov udeo

Komercijalni kompjuteri u Engleskoj vuku deo svojih korena sa prilično neobičnog mesta Inventivna ideja da se počne sa proizvodnjom računara za automatizaciju kancelarijskog poslovanje potekla je od kompanija Lajons (Lyons) koja je posedovala niz malih prodavnica prehrambenih proizvoda i čaja Ova kompanija imala je dugu tradiciju inovacije u poslovnim metodama Uvela je kalkulatorske mašine u svoje prodavnice još davne 1896 godine a od tridesetih godine ovog veka eksperimentisali su se mikrofilmovima na kojima su bile zabeležene informacije o svim finan-



*Nezasluženi primat UNIVAC nije kako se to često misli prva mašina na svetu sa unutrašnjim programom primat zapravo pripada Englezima*

### Prvi mikroprocesor

*Prvi mikroprocesor 4004 razradio je Intel krajem 1970 godina po projektu Marsijana Hofa (Marcian Hoff), koji je želeći da smanji broj čipova u kalkulatorima došao na genijalnu ideju da načini čip koji bi bio nalik na centralna procesore velikih računara To je bio četvorobitni procesor koji je obrađivao podatka u blokovima od po 4 binarne cifra Sa 2250 tranzistora mogao je da obavlja 60 000 operacija u sekundi*

sijskim transakcijama Svakih nekoliko godina Lajons je slao svoje planera u inostranstvo radi proučavanja novih metoda u poslovanju pa su tako dva njegova službanika 1947 godina otišla u Ameriku da na delu vide elektronske mozgova koji su bili glavni hit tog vremena Najznačajnija informacija koju je Lajons dobio sa tog studijskog putovanja bila je da se elektronski mozgovi proizvode i mnogo bliže — u Kembridžu

Na osnovu analize koja je pokazivala da bi u sopstveni kompjutar trebalo investirati 100 000 funti ali da bi njegova primena godišnja uštedela 50 000 funti kompanija je u oktobru 1947 godina započela rad na projaktu sopstvenog računara Ono što izaziva posebno poštovanje je činjenica da se kompanija odlučila za ovaj projekat još u vrame kada je na Kembridžu računar EDSAC bio u projektu Za početak stipendirali su se 3 000 funti rad na EDSAC-u i time bitno ubrzali njegov završetak tako da je on već 1949 sa uspehom obavio svoj prvi posao — računanje tablice prostih brojeva

Lajons je analizirao probleme koje bi njegov kompjutar trebalo da rešava i skicirao rutine koje su mu za to neophodne Ova studije predstavljala su osnovu za prve programa i pomogla da se definišu hardvarska rešenja Već na samom početku bilo je jasno da se poslovni računar mora suštinski razlikovati od univerzitetske mašine koje se upravo razvijala Dok je EDSAC bio projektovan za izvršavanje komplikovanih i dugotrajnih matematičkih izračunavanja pri čemu je sa ulaza zahtevao samo nekoliko podataka poslovni računar trebalo je da rešava sasvim drugačije probleme Matematička operacije u poslovnim obradama su veoma jednostavne ali se moraju vršiti nad mnoštvom podataka

LEO (the Lyons Electronic Office) je počeo da se koristi od 9 februara 1954 godine za obračun nedeljnih plata za 1 700 zaposlenih LEO je za 1 5 sekundi obavljao posao za koji je službaniku bilo neophodno najmanje 8 minuta

U Lajonsu su ubrzo uviđeli da je potrebno imati više ovakvih računara a postojao je i ogroman interes drugih firmi, pa se Lajons počeo baviti i proizvodnjom i prodajom kompjutera Tako je iskustvo stečeno projektovanjem LEA iskorišćeno za izgradnju čitave serije poboljšanih verzija ovog računara koji su se uspehom korišćeni u biznisu vladinim organizacijama i industriji Zahvaljujući kvalitetnim računarima koje je nudila domaća industrija Velika Britanija odolela je invaziji IBM-a i sedam patuljaka, koji i danas žare i pale na evropskom tržištu komercijalnih računara

* * *

Savremenu industriju velikih računarskih sistema danas vode IBM i sedam patuljaka kojima jedino Japan može da ugrabi deo tržišta IBM je finansijski tako ojačao da godišnja obrne preko 20 milijardi dolara Patuljci – Burroughs NCR Control Data Sperry Rand, DEC Honeywell i Hewlett-Packard, svaki sa godišnjim obrtom kapitala od oko 3 milijarde dolara — imaju takvu finansijsku moć da je teško zamisliti da će u skoroj budućnosti neka nova firma moći da im ugrozi tržište Međutim lekcije iz računarske istorije kazuju da su kvalitetno novi proizvodi i nove tehnologije dolazili sa neočekivanih mesta i takvom brzinom da su „veliki' jedva uspevali da ih prate Nisu li primeri personalnih računara i Apple-a, firme koja je imala najbrži rast u istoriji američke berze najbolja potvrda ova teze U tome se, možda, krije i naša šansa da uđemo u istoriju kompjutera.

***Nevenka Spalević***
***Branko Đaković***

# Dejanove pitalice

## Nagradna premetaljka

Naš treći nagradni zadatak je pokazao da su čitaoci „Računara" pošteni radni ljudi — nije ih mnogo interesovala godina sa najmanjim brojem radnih dana.

Obzirom da smo u predviđenom roku dobili samo tri potpuno tačna rešenja (stiglo je samo pedesetak odgovora, možda zato što uz zadatak tehničkom greškom nije objavljen nagradni kupon), nije nam bilo teško da rasporedimo nagrade. Prvu nagradu od 10 000 dinara je dobio **Frano Gospodnetić** iz Zagreba, drugu (5 000 din) **Zlatko Tratar** iz Ljubljane a treću (3 000 dinara) **Geza Berdalović** iz Bilja.

Šta je najviše namučilo rešavače? Pre svega nepoznavanje kalendara: po Gregorijanskom („novom") kalendaru godine kojima se završavaju vekovi su prestupne samo ako je njihov broj deljiv sa 400, što znači da je 2000. godina prestupna ali da 2100, 2200 i 2300 nisu. Kalendar se dakle ponavlja svakih 400 godina, što je bio osnovni razlog da u postavci problema pomenemo razdoblje 1986—2386. Sledeći je problem bio što je za rešavanje zadatka **zaista** bio potreban računar: bez njega se teško može obraditi veliki broj godina tako da rezultati budu pouzdani. Dobili smo dvadesetak pisama u kojima su navedeni programi koji bi trebalo da reše problem ali ne i rezultati izvršavanja tih programa — izgleda da u našim krajevima i dalje ima više programera nego računara. Kao tačna rešenja **priznajemo** pisma u kojima nije naveden postupak rešavanja (takva pisma, doduše, retko konkurišu za prvu nagradu) ali **ne priznajemo** pisma sa programima u kojima nije naveden konkretan i precizan odgovor na postavljeno pitanje! Poželjno je da taj odgovor ne bude zakopan u sredinu (nečitko pisanog) teksta.

Na slici su data dva rešenja problema: jedno je dalo Frano Gospodnetić pisano na kod nas malo poznatom jeziku C i izvršavano na nekoj CP/M mašini, a drugo primer umešnosti Zlatka Tratara na Sharpu M7700.

Pokazuje se da najviše neradnih dana (po 114) imaju prestupne godine kod kojih prvi januar pada u nedelju. To su 2012, 2040, 2068, 2096, 2108, 2136, 2164, 2192, 2232, 2260, 2288, 2328, 2356 i 2384. Do prve (naj)srećnije Nove godine će, dakle, proteći još čitavih 26 leta.

```
100 REM PROB/2
110 REM PROGRAM ZA IZRACUN LETA Z NAJVECJIM STEVILOM DELA PROSTIH DNI
120 REM PROGRAMER: ZLATKO TRATAR DATUM: 08.01.88 RACUNALNIK: SHARP MZ-700
130 :
140 :
150 REM *** PRIPRAVA ***
160 DIM PR(3,1),PD(1,8)
165 FOR I=0 TO 3: READ PR(I,0),PR(I,1): NEXT I
170 FOR I=0 TO 8: READ PD(0,I),PD(1,I): NEXT I
180 MD=0: TD=0
190 :
200 :
210 REM *** G L A V N I   P R O G R A M ***
220 FOR LL=1986 TO 2385
230 TD=0
240 REM *** IZRACUN STEVILA DNI V TEKOCEM LETU ***
250 X1=INT(LL/100): X2=LL/100: X3=INT(LL/400): X4=LL/400
260 IF (X1=X2)*(X3<>X4) THEN DL=365: GOTO 280
270 DL=366-SGN(LL-INT(LL/4)*4)
275 :
280 REM *** IZRACUN STEVILA DELA PROSTIH DNI V LETU ***
290 FOR I=0 TO 3
300 DD=PR(I,0): MM=PR(I,1)
310 REM *** DOLOCITEV DNEVA V TEDNU - ZELLERJEVA KONGRUENCA ***
320 L2=LL
330 IF MM>2 THEN MM=MM-2: GOTO 340
335 MM=MM+10: L2=L2-1
340 L1=INT(L2/100): L2=L2-INT(L2/100)*100
350 DN=DD+INT((13*MM-1)/5)+INT(2*L1/4)+INT(5*L2/4)
360 DN=DN-INT(DN/7)*7
370 REM *** IZRACUN STEVILA PRAZNICNIH DNI V TEKOCEM LETU ***
380 TD=TD+PD(SGN(I),DN)
382 IF I<>0 THEN 450
384 IF ((DL=365)*(DN=1))+((DL=366)*(DN=0)) THEN TD=TD+1
386 IF DN=2 THEN TD=TD-1
400 REM *** IZRACUN STEVILA SOBOT IN NEDELJ V TEKOCEM LETU ***
420 X1=DL+DN-8: TD=TD+INT(X1/7)+SGN(X1-INT(X1/7)*7)
430 IF DN<>0 THEN X1=X1: GOTO 440
435 X1=DL
440 TD=TD+INT(X1/7)+SGN(X1-INT(X1/7)*7)
450 NEXT I
460 :
470 REM *** HRAMBA NAJVECJEGA STEVILA DELA PROSTIH DNI ***
480 IF TD>MD THEN MD=TD: ML=LL
500 NEXT LL
510
520 :
530 REM *** IZPIS ***
540 PRINT/P "PRVO LETO=";ML;"  STEVILO DNI=";MD
550 END
570 :
580 REM *** DATUMI PRAZNICNIH DNI ***
590 DATA 4,7, 1,1, 1,5, 29,11
600 REM *** STEVILO PRAZNICNIH DNI IZVEN SOBOT IN NEDELJ, CE PADE PRAZNIK NA
610 REM   NED  PON  TOR  SRE  CET  PET  SOB
620 DATA 1,2, 1,2, 1,2, 1,2, 1,2, 1,1, 0,1: REM ENODNEVNI/DVODNEVNI
PRVO LETO= 2012  STEVILO DNI= 114
```

```
/*****************************
 * RACUNARI C       Narodni dani *
 *                               *
 *      Frano GOSPODNETIC,       *
 *      Vinkoviceva 8-B,         *
 *          Zagreb               *
 *        05/01/1988             *
 *                               *
 * BDS SOFTWARE C Compiler, v1.5 *
 *****************************/

#define CLS      0x1A
#define SUNDAY   2
#define IS_LEAP_YEAR(year)   year % 4 == 0 && year % 100 != 0
                                         || year % 400 == 0

main ()
  {
     int year, New_Year;

         putchar (CLS);
         New_Year = 1986 % 7;

         for (year = 1400; year <= 2385; year++)
           {
             if ((IS_LEAP_YEAR(year)) && New_Year == SUNDAY)
                printf ("%4d\n", year);

             if (IS_LEAP_YEAR(year))
                New_Year += 2;
                else
                New_Year++;
             if (New_Year > 6)
                New_Year -= 7;
```

Novi nagradni zadatak se zasniva na igri „Reverse" koja je u svoje vreme bila čest gost memorije džepnih programabilnih kalkulatora. Dat vam je neima tekst „I_U_VARNIČARI" koji treba transformisati u „RAČUNARI_I_VI" (i donja crta koju smo koristili u obe reči je karakter ravnopravan svakom slovu) primenom što manjeg broja dozvoljenih transformacija. Dozvoljena je jedino transformacija kojom menjate redosled prvih nekoliko slova date reči dok se njen rep ne menja. Ako, na primer, promenite redosled prvih pet slova u početnoj reči „I-U-VARNIČARI", dobićete „V_U_IARNIČARI", a ako zatim u toj reči promenite redosled prvih 8 slova, dobijate „NRAI_U_VIČARI". Korak po korak dobićete tekst „RAČUNAR_I_VI". Zatim na list papira ispišite sve korake, popunite kupon koji ovde dajemo i sve to pre 20. marta pošaljite na adresu „Računari — za Dejanove pitalice", Bulevar vojvode Mišića 17, Beograd. Nagrade od 10 000, 5 000 i 3 000 dinara čekaju rešavače koji do rešenja dođu u minimalnom broju poteza, ukoliko ih bude više, odlučiće žreb.

Tekst 'I_U_VARNICARI' se moze transformisati u 'RACUNARI_I_VI' primenom najmanje ______________ legalnih transformacija

Ime i prezime ______________

Adresa ______________

Mesto (________) ______________

# Holandska zastava

***Čuveni Dijkstra je 1976. godine postavio zadatak koji je nazvao „problem bojenja holandske zastave" Iako deluje krajnje jednostavno i lako razumljivo zadatak još do danas nije dobio najelegantnije rešenje U čemu je to problem sa holandskom zastavom i kako ga je dva puta rešio sam Dijkstra a zatim i, nešto bolje, Mejer***

Neka je dat niz od N elemenata od kojih je svaki obojen u jednu od tri boje — crvenu belu ili plavu Problem je kako pregrupisati elementa niza tako da svi crveni elementi budu na početku beli u sredini a plavi na kraju Sada je verovatno svima jasno zašto se zadatak zove problem holandske zastave

## Primitivne operacije

Pri pravljenju zastave Dijkstra dozvoljava korišćenje dve primitivne operacije

buck(i) — funkcija koja očitava boju i-tog elementa niza

swap(i j) — procedura koja menja i-ti i j-ti element niza tj i-ti premešta na mesto j-tog, a j-ti na mesto i-tog

Dodatni uslovi za program su da

1) mora da funkcioniše za sve moguće početne raspodele elemenata,

2) primena buck operacije dvaput na isti element nije dozvoljena

Dijkstra je takođe naglasio da od programa približne složenosti više ceni onaj sa manjim brojem zamena (swap) Sva rešenja koja ćemo prikazati razlikuju četiri klase elemenata niza — crvene bele plave i elementa čiju boju još nismo pročitali Elementi su grupisani u četiri regije a informacije o granicama između tih oblasti čuvaju se u tri promenljive **r, w** i **b** kao na slici

| 1 | | | N |
|---|---|---|---|
| crveni | neispitani | beli | plavi |
| r | | w | b |

odnosno za

1<=k<r k-ti element je crven (red),

r<=k<=w k-ti element nije ispitan tj još ne znamo koje je boje

w<k<=b k-ti element je beo (white)

b<k<=N k-ti element je plav (blue)

## Tri pristupa

PRVI PRISTUP je da ispitujemo po jedan element iz klase neispitanih i smeštamo ga u odgovarajuću klasu Pri tome naravno ažuriramo granice oblasti Paskal procedura ovog sistema algoritma data je na slici 1

DRUGI PRISTUP je poboljšanje prethodnog koje je dao autor problema Primetimo da prethodni algoritam pravi N zamena i u slučaju kada su inicijalno svi elementi niza crveni što je vrlo neracionalno Nova ideja je da

Neispitanu oblast možemo skanirati sleva nadesno (koristeći promenljivu **r**) Kad naiđemo na element koji nije crven počinjemo skaniranje neispitane oblasti sdesna ulevo (koristeći promenljivu **w**) Pri tome preskačemo sve bele elemente sve dok na naiđemo na neki koji nije beo Ako je taj element plav zamenjujemo ga elementom na koji ukazuje **b** i nastavljamo skaniranje Kada promenljivom **w** naiđemo na crveni element zamenjujemo ga elementom na koji ukazuje **r** (dakle nije crven) Osim toga ako

```
PROCEDURE holandskazastava3I
 VAR r,w,b Integer;
     colr,colw color;
BEGIN
 r:=1; w:=N; b:=N;
 WHILE w>=r DO
  BEGIN
   colr:=buck(r);
   WHILE (colr = red) AND (w>r) DO
    BEGIN
     r:=r+1;
     colr:=buck(r)
    END;
   IF r<w THEN
    BEGIN
     colw:=buck(w);
     WHILE (colw <> red) AND (w>r+1) DO
      BEGIN
       IF colw = blue
        THEN
         BEGIN
          swap(w,b);
          b:=b-1
         END;
       w:=w-1;
       colw:=buck(w)
      END;
     IF colw [illegible]
      THEN
       BEGIN
        swap(w,b);
        b:=b-1
       END;
     IF [illegible] red) OR (colr <> white)
      THEN
       swap(r,w);
    END;
   IF c[illegible] r
    THEN
     BEGIN
      swap(w,b);
      b:=b-1
     END;
    r:=r+1;
    w:=w-1
  END
END {holandskazastava3};

PROCEDURE holandskazastava1;
 VAR r,w,b: integer;
BEGIN
 r:=1; w:=N; b:=N;
 WHILE w>=r DO
  CASE buck(w) OF
   red   : BEGIN
            swap(r,w);
            r:=r+1
           END;
   white : w:=w-1;
   blue    BEGIN
            swap(w,b);
            w:=w-1;
            b:=b-1
           END
  END
END {holandskazastava1};

PROCEDURE holandskazastava2;
 VAR r,w,b: integer;
     colr,colw: color;
BEGIN
 r:=1; w:=N; b:=N;
 WHILE w>=r DO
  BEGIN
   colr:=buck(r);
   WHILE (colr = red) AND (w>r) DO
    BEGIN
     r:=r+1;
     colr:=buck(r)
    END;
   IF r<w THEN
    BEGIN
     colw:=buck(w);
     WHILE (colw = white) AND (w>r+1) DO
      BEGIN
       w:=w-1;
       colw:=buck(w)
      END;
     CASE colw OF
      red   : BEGIN
               swap(r,w);
               r:=r+1
              END;
      white : w:=w-1;
      blue  : BEGIN
               swap(w,b);
               w:=w-1;
               swap(r,w);
               b:=b-1
              END
     END
    END {if r<w};
   IF colr = blue
    THEN
     BEGIN
      swap(w,b);
      b:=b-1
     END;
   w:=w-1
  END
END {holandskazastava2};
```

(I) granicu r crvene oblasti pomeramo desno što je više moguće bez zamena

(II) slično granicu w bele oblasti pomeramo u levo što je više moguće bez zamena

(III) možda bi bilo korisno da istovremeno ispitujemo po dva elementa niza umesto jednog (granicu crvene i neispitane oblasti i granicu neispitane i bele oblasti),

Ako ovo implementiramo element na koji ukazuje r pointer (ili kako bi to rekli naši profesori pokazivač) biće ili beo ili plav a element na koji ukazuje w ili crven ili plav Sve dok je r<w−1 postoje četiri klase Paskal procedura ovog algoritma data je na slici 2

Majerovo rešenje prevazilazi nedostatak prethodna dva Naime u njima postoje situacije u kojima premeštamo element čiju boju ne znamo što je neracionalno jer postoje šanse da je on već u dobroj klasi je taj bivši r plav zamenjujemo ga elementom na koji pokazuje **b** Paskal procedura ovog algoritma data je na slici 3

## Još bolje

Ovaj algoritam se može i dalje poboljšavati Umesto skaniranja ulevo što je moguće dalje bez zamena granicu pomeramo ulevo dok se na pojavi crveni element i pri tom preskačemo bele a zamenjujemo plave elemente ili sve tri pojavljivanja swap(w b) zamenimo sa if w b then swap(b w) Zanimljiva je i ideja da se istovremeno posmatraju više od dva elementa istovremeno Poboljšanja međutim treba da budu i korisna — komplikacije u programu moraju da imaju svoje praktično opravdanje! Pokušajte i vi sa nekim poboljšanjima Još uvek nije rečena poslednja reč

***Zoran Obradović***

# CP/M

CP/M je univerzalni operativni sistem implementiran u *raznim verzijama* na mnogim ličnim računarima različitih proizvođača. Izraz *u raznim verzijama* verovatno u vama budi opravdanu sumnju u kojoj su meri te razne verzije međusobno kompatibilne? Odgovor na ovakvo pitanje nas vodi u rane sedamdesete godine doba velikog entuzijazma u razvoju kućnih računara. Gari Kildal (Gary Kildall) profesor mornaričkih postdiplomskih studija u Kaliforniji i Intelov softverski konsultant tih je dana od na otpadu nabavljenih komponenata i Intelovih čipova van specifikacije sastavljao svoj prvi sasvim lični računar koji je, naravno, trebalo opremiti nekakvim softverom. Pošto je bilo skoro nemoguće asemblirati iole složeniji operativni sistem „na ruke" Gari je posle radnog vremena krišom koristio Intelov razvojni sistem da prevede najnoviju verziju svoga programa sa PL/M-a na mašinac i ugradi ga u eprom koji bi iste večeri isprobavao kod kuće. I pored ovako teških uslova operativni sistem Garijevog kompjutera je polako napredovao ka konačnoj verziji koja je nazvana CP/M 1.3 (CP/M je kao što svakako znate skraćenica od Control Program/Monitor).

## *Zakrpa na zakrpu*

Geri Kildal je sasvim prirodno, ponudio svoje remek-delo Intelu koji je verujući da su kućni računari interesantni samo za malo i sumnjivo tržište amatera, odbio da otkupi CP/M i tako napravio jednu od dve najveće gluposti u istoriji svoje inače veoma uspešne egzistencije (druga glupost je što nije našao zajednički jezik sa konstruktorima 8080 koji su se zbog toga odvojili i osnovali Zilog). Garija ovo odbijanje nije previše obeshrabrilo pa je zajedno sa ženom Doroti koja se bavila marketingom osnovao danas čuvenu firmu Digital Research. Početni kapital sakupljen od akcionara je bio dovoljan za kupovinu razvojnog sistema i opreme za programiranje eproma (tada veoma skupa) pa je Geri bez mnogo problema ispravio neke bagove CP/M-a 1.3 i tako dobio verziju 1.4 koja je i danas u primeni na nekim kompjuterima. Čak je i početna verzija 1.3 prodata firmi Cromemco danas u životu pod imenom Cromemco Disc Operating System — CDOS.

Proširivanjem memorija kućnih računara krajem sedamdesetih godina CP/M 1.4 je bivao sve zastareliji. Ovaj program je, naime, koncipiran tako da zauzima što manje mesta u memoriji pa mu je bila neophodna česta komunikacija sa diskom koja je odnosila previše vremena. S druge strane kapaciteti disketa su bili prilično mali pa se sa dve drajva neki poslovi jednostavno nisu mogli obaviti. CP/M 2.2 je jasno lako prevazišao prvi problem: bitan deo programa je premešten u memoriju. Da bi se prevazišao problem nedovoljnog kapaciteta flopi diskova Digital je došao na ideju da softverski podrži čak 16 disk jedinica koje bi bile povezane sa centralnim procesorom. U doba današnjeg rapidnog pada cena masivnih (hard) diskova ovakvo solucija može da izgleda veoma smešno: zamislite vaš sto pretrpan drajvovima i sebe kako žonglirate sa disketama! Pa ipak pre nekoliko godina se samo na ovaj način mogle skladištiti dovoljne količine informacija za ozbiljnije poslovne primene. Fleksibilne izmene CP/M-a su učinile da ovaj operativni sistem krajem 1979 godine koristi preko 1.5 milion vlasnika računara!

Dalji napredak hardvera je omogućio ozbiljno razmišljenje o personalnim računarima koji bi istovremeno obradivali više programa ili u složenijem slučaju opsluživali više korisnika. Tako je nastao MP/M (*Multi Programming Monitor control program*) operativni sistem koji nikada neće postići naročitu popularnost. Praktično istovremeno IBM je odlučio da svoj danas neverovatno popularni računar PC zasnovan na Intelovom mikroprocesoru 8080 opremi operativnim sistemom koji liči na CP/M. Iako je Digital planirao da pripremi operativni sistem koji bi nazvao CP/M 86, firma Seattle Computer Products ga je preduhitrila napisavši operativni sistem koji je najpre nazvan 86-DOS a onda malo zvučnije QDOS. IBM je svoj PC opremio MS DOS-om hibridom Microsoftovog disk operativnog sistema i 86-DOS-a. MS DOS je na žalost nekompatibilan sa CP/M-om iako su naredbe slične, format informacija na disketama je bitno promenjen što znači da je prelazak sa nekog CP/M računara na IBM PC prilično bolan.

Digital je za to vreme pripremao CP/M 3.0 koji je u poslednjem trenutku (po cenu preštampavanja čitave dokumentacije) preimenovan u CP/M PLUS jer su stručnjaci za marketing smatrali da se na taj način snažnije podvlači unapređenja tada već izuzetno popularnog operativnog sistema. Digital se osim toga neprakidno trudi da prilagodava CP/M svim novim mikroprocesorima koji se pojavljuju na tržištu pa je tako nastao CP/M-68K (za Motorolu 68000) i CP/M-Z8K (za Zilog Z8000). Implementacija na moćnijim procesorima su međutim iznosila na videlo neke loše osobine CP/M-a pa je ovaj operativni sistem morao da bude dopunjavan dodatnim sistemskim softverom, kao što je GEM na atariju 520 ST. Na slici je prikazano stablo raznih verzija operativnog sistema o kojem govorimo.

slika 1

CP/M-80 version 1.3
CP/M-80 version 1.4
CDOS (Cromemco)
CP/M-80 version 2.2
MP/[illegible]
CP/M-PLUS
CP/M-86
MP/M-86
CP/M-68K
CP/M-Z8K
Concurrent CP/M 86

## *Metuzalem sa pedigreom*

O dobrim stranama CP/M-a se mnogo piše: radi se o standardnom operativnom sistemu sa ogromnom softverskom podrškom, opsežnom dokumentacijom i mnoštvom knjiga za početnike. CP/M je, osim toga otvoren operativni sistem koji se uz malo veštine u pisanju mašinskih programa može prilagodavati potrebama korisnika. Hardverski zahtevi CP/M-a su prilično skromni, što znači da se ovaj operativni sistem može instalirati uz umerena novčana ulaganja: računar sa Z80, 64 Kb RAM-a, dve disk jedinice i monitorom danas može da se nabavi za manje od 600 funti što uopšte nije mnogo za poslovne primene čak ni u našim uslovima.

Sve loše osobine CP/M-a se mogu iskazati jednom rečju: zastareo. CP/M se nije razvija već petnaestak godina za to vreme je bezbroj puta krpljen da bi udovoljio zahtevima novih generacija hardvera. Ukoliko na primer aktivirate neki CP/M kompjuter i otkucate DIT a zatim primetivši da ste pogrešili pritisnete DELETE nemojte se previše iznenaditi ako se na ekranu pojavi DITT. Početne verzije CP/M-a su, naime predviđale rad sa konzolnim terminalom (teleprinterom) koji jednostavno ne može da vrati glavu za jedno mesto unazad, brisanje simbola je signalizirano njegovim ponavljanjem! Za video terminale koji su danas u isključivoj upotrebi ovakav rad predstavlja običnu besmislicu, ali se ipak održao u nekim implementacijama.

Zastarelost CP/M-a se daleko više primećuje u radu sa grafikom i zvukom. Za tvorce CP/M-a je bilo nezamislivo da se kompjuterska grafika na malim računarima može koristiti i za šta drugo osim za igre. Igre su, mislilo se, prilično nepotrebne jednom poslovnom operativnom sistemu. Zaključak: CP/M nema grafiku. Nikakvu!

Pokazalo se međutim da se igre mogu pisati i sa rezolucijom 80*24 ili 80*32 s tim

***Interesovanje koje je izazvao naš napis „Prvi gem za Tripos“ u prošlim „Računarima“ nas je podstakao da započnemo seriju tekstova o popularnim operativnim sistemima ličnih kompjutera. Teško bi bilo pronaći ljubitelja računara koji, kada se pomene operativni sistem, neće pomisliti na CP/M, program o kome je napisano mnogo knjiga i tekstova (čak i na našem jeziku!). I pored toga, verujemo da većini naših čitalaca nedostaje potpuna i relevantna informacija iz koje će naučiti kako se razgovara sa kompjuterom na jeziku koji je firma Digital Research nazvala CP/M.***

što će svemirci biti zamenjeni zvezdicama pozitivci uzvičnikom a ograda koja ih štiti iksevima Ne izgleda baš mnogo veselo pogotovu zato što nema zvuka ali je sasvim upotrebljivo! Bilo kakvi programi koji koriste grafiku u poslovne svrhe (dijagrami histogrami torta dijagrami i slični indikatori) su međutim u startu neostvarljivi! Mnogi ozbiljni vlasnici računara bi se danas rado bavili kompjutarskim simulacijama animacijom i sličnim područjima ali je za njih potrebno tražiti drugi operativni sistem Umesto celovitog rešenja tvorci raznih CP/M kompjutera su pripremali uslužne mašinske programe za rad sa grafikom koji su bili prilično teški za upotrebu nefleksibilni slabo dokumentovani i što je najgore nestandardizovani GEM grafički operativni sistem koji smo opisali u prošlim Računarima predstavlja ozbiljno proširenje CP/M-a (i MS DOS-a) koje tek treba da se pokaže na tržištu U međuvremenu CP/M je operativni sistem upotrebljiv jedino za rad sa slovima

Međusobna kompatibilnost raznih CP/M računara je takođe prilično sumnjiva Minimalan uslov za kompatibilnost je format diskova — ako ste na poslu snimili neke podatke na disketu *neophodno* vam je da možete da ih pročitate i na svom kućnom računaru CP/M je međutim često instaliran kao drugi operativni sistem Apple je na primer radio pod Apple DOS-om pa se dokupljivanjem jednog Z80 pretvarao u CP/M mašinu Jasno je da se uz Z80 obično dobijao minimum hardvera pa se o komunikaciji sa diskovima i dalje brinuo 6502 format disketa je i dalje bio Appleov!

## Kompatibilnost bez pokrića

Kada se problem kompatibilnosti disketa nekako reši požalećemo i kompatabilnost softvera Jasno je da programi za dva kompjutera koji koriste razne mikroprocesore ne mogu ni slučajno da budu kompatibilni ni jedan od čuvenih takst procesora koji savršeno rade na partneru neće moći da se izvršava na atariju 520 ST ! Moguće je da će u budućnosti zahvaljujući ogromnoj razlici u brzini rada biti napisan softverski emulator pomoću koga će Motorola 68000 savršeno imitirati Z80 pa će dobar deo softvera biti direktno upotrebljiv Moguće je isto tako da se softverske firme ljuto usprotive proizvodnji ovakvog emulatora CP/M programi su široko rasprostranjeni (= mnogo se piratuju) pa se proizvođačima softvera daleko više isplati da svoje remek-dela sami prodaju u više sličnih varijanti Ne sme vas čak začuditi ni ako program sa partnera ne radi na drugim CP/M mašinama za Z80 autorima programa se više isplati da posebno prilagođavaju softver svakom računaru i tako umnožavaju svoje prihode!

CP/M-u se najzad zamera da je slabo opremljen naredbama za rad sa periferijom i da su ove naredbe nedovoljno standardizovane Ovakvim zamerkama po našem mišljenju nema mnogo mesta naredbe za rad sa diskom i štampačem su integralni deo CP/M-a dok ploterima optičkim perima i drugim hardverom i onako uglavnom upravljaju aplikacioni programi U ovom trenutku je uostalom specijalna grafička oprema prilično malo dok će u danima kada ona postane dovoljno jeftina da bi bila masovna CP/M verovatno otići u penziju ili biti dopunjen nekim specijalizovanim operativnim sistemom kao što je GEM

Govorili smo dakle o nastanku CP/M-a njegovom razvoju dobrim i lošim stranama Došlo je vreme da pređemo na konkretnije stvari i opišemo standardne naredbe CP/M-a pri čemu ćemo uglavnom imati u vidu CP/M 2 2 i CP/M PLUS dok će MP/M i druge varijante ovoga operativnog sistema biti pominjane samo u posebnim slučajevima

## Osnovne naredbe

Pretpostavimo da su na vaš računar koji radi pod CP/M-om priključene dva dvostrane (ili što je sa funkcionalnog aspekta isto), četiri jednostrane disk jedinice označene slovima A B, C i D U svakom drajvu se jasno, nalazi po jedna disketa spremna za rad CP/M dok mu drukčije ne naredite smatra da ći se sve naredbe za rad sa diskom odnositi na drajv A pa ispisuje prompt A> znak da očekuje instrukciju Prva naredba koju ćete otkucati će verovatno, biti DIR, skraćenica od *directory* zahtevate od računara da prikaže imena svih programa koji su upisani na sistemsku disketu Disketa će se zavrteti i na ekranu pojaviti nešto poput slike 2

Na disketi se, dakle nalazi osam programa Prvi se zove STAT COM drugi MBASIC4 COM treći BACKUP COM i tako dalje sve do poslednjeg fajla sa čudnim imenom PIP COM Vidimo da se kompletan naziv programa sastoji od imena (npr STAT, MBASIC4 PIP i slično) koje može sadržati najviše osam slova i oznake njegovog tipa (u našem slučaju su svi programi tipa COM) koja se sastoji od najviše tri slova Dok je svakom čitaocu ovih redova jasno da program mora imati unikatno ime navođenje oznake tipa može da nam izgleda kao nepotrebno usložnjavanje Pokazalo se međutim da korisnici CP/M-a često pišu programe na višim jezicima koji se kompajliraju na primer paskalu ili fortranu Kada naredite računaru da prevede program PROBA FOR na mašinski jezik objektni kod će morati da bude smešten u neku datoteku koju bi bilo logično nazvati PROBA OBJ (mogli bismo je naravno nazvati i PROBA1 ili PROBAOBJ ali bi se na taj način znatno gubilo na preglednosti i povećavala verovatnoća nastanka zabune) Zbog toga su standardima CP/M-a predloženi osnovni tipovi programa koji su popisani na slici 3 Nama, naravno preprека da i sami kreirate datoteke nekog posebnog tipa (na primer IZVESTAJ JUN ili REŠENJA JAN) pri čemu morate da se pridržavate zlatnih pravila u okviru oznake tipe programa smeju da se pišu **isključivo** slova i cifre dok se u okviru imena programa mogu

slika 2

```
A>DIR
A:STAT     COM     MBASIC4    COM     BACKUP    COM     MBASIC5    COM
A:CBASZ    COM     CRUN2      COM     XREF      COM     PIP        COM
A>
```

slika 3

| | |
|---|---|
| .ASM | Izvorni (source) program na asembleru. |
| .BAK | Backup neke datoteke |
| .BAS | Izvorni program na bejziku |
| CMD | Objektni relokatibilni program (CP/M 86) |
| COM | Izvršni program |
| DAT | Podaci |
| DOC | Datoteka sa tekstom - dokumentacijom. |
| .HEX | Program u Intelov HEX formatu. |
| INT | Delimično prevedeni bejzik program (CBASIC). |
| LIB | Biblioteka potprograma |
| MAC | Izvorni program sa makro naredbama. |
| .PRL | Program koji se može izvršiti na bilo kojoj strani memorije (MP/M). |
| .PRN | Izveštaj koji asembler priprema za štampanje |
| SUB | Procedura koja se prosleđuje SUBMIT programu. |
| TXT | Datoteka sa tekstom |
| $$$ | Privremena datoteka koju treba obrisati na kraju sesije |

koristiti! I neki specijalni znaci ali ne upitnik zvezdica zagrade (uglaste obične i znaci veće i manje) tačke zarez kose crte, znak jednakosti i dvotačka U tom su smislu korektne imena fajlova DAN-PRVI CDM CPM RAC LUJ XIV GALAKS 1 8271 101 i DH! ali ne I CP/M RUN CPM A-B GALAKSIJA 1 ili JE ) FAJL BAS

## Direktorijum na dlanu

Komanda DIR će na ekranu prikazati imena i tipove svih datoteka osim programa tipa SYS, da biste dobili spisak ovih programa moraćete da otkucate DIRSYS (ili DIR* SYS ukoliko ne koristite CP/M PLUS) Čemu ovo skrivanje? Pokazalo se da neiskusne korisnike koji bi želeli da se služe računarom ne puštajući se u programiranje ili operativni sistem može da zbuni pojava mnoštva programa na disketi pa je ostavljena mogućnost da se neki od njih sakriju dok će se u direktorijumu nalaziti samo programi koje ima smisla samostalno startovati

Ako umesto DIR, napišete DIR PRDBA-BAS računar će prikazati samo programe koji se zovu PRDBA BAS Na disketi jasno može da postoji samo jedan takav program pa će se ova opcija prilično retko koristiti Sa druge strane često ćemo u imenu koje navodimo iza DIR koristiti džoker znake upitnik i zvezdicu Upitnik kao i obično zamenjuje bilo koji znak a zvezdica bilo koju grupu znakova pa će tako DIR* BAS ispisati naslove svih bejzik programa, DIR PRDBA * pronaći programe svih tipova koji se zovu PRDBA dok će DIR ??AB?? COM dati imena programa koji se na primer zovu TABLA COM STABLO COM i DIABLD COM Ukoliko poželimo da proverimo postojanje nekih programa na drugim drajvovima koristićemo DIR B * * DIR C PROBA * i tome slično

Pošto ste isprobali naredbu DIR pokušajte da otkucate DIR FULL (samo ako koristite CP/M PLUS) i videćete nešto poput slike 4 Uz imena programa sa primećujemo na disketi čuvaju i neki drugi podaci kao što je veličina u blokovima i bajtovima podatak o zaštiti programa i kod nekih verzija datum kreiranja Sa DIR [EXCLUDE] možemo da dobijemo imena svih datoteka osim onih koji odgovaraju nekoj specifikaciji sa DIR [RO] da prikažemo imena svih programa koji su otvoreno isključivo za čitanje i tako dalje

Pre nego što nastavimo sa opisivanjem naredbi CP/M-a treba da razmotrimo pitanje zaštite programa koju smo već dva puta pomenuli CP/M se u globalu ne bavi zaštitom podataka pošto je namenjen prevashodno računarima koje će koristiti jedna jedina osoba Pa ipak ponekad će ta osoba poželeti da zaštiti neke programe od samoga sebe, to jest da se zaštiti od grešaka koje mogu da dovedu do gubitka vrednih podataka Otkucajte dakle SET A *.COM [RO SYS] Na taj ćete način ova programe tipa COM proglasiti zatvorenim za upis (bilo kakav pokušaj da se neki od njih menja ili briše će izazvati poruku o grešci) i uzgred zatražiti od računara da njihova naslove ne prikazuje kada korisnik otkuca DIR Ako docnije poželite da promenite neki od ovih programa možete da otkucate SET A * * [RW DIR] (RO dolazi od *read only* a RW od

```
slika 4

A>DIR [FULL]

Scanning Directory
Sorting Directory

Directory for Drive A:  User 0

   Name      Bytes  Recs  Attributes        Name       Bytes  Recs  Attributes
 -----  ---  -----  ----  --- -- --        ---------- -----  ----  ----------
CPM     SYS    18k   138  Sys  RO          DATE    COM    3k    22  Dir  RW
DEVICE  COM     8k    58  Dir  RW          DIR     COM   15k   114  Dir  RW
ED      COM    10k    73  Dir  RW          ERASE   COM    4k    29  Dir  RW
HELP    COM     7k    58  Dir  RW          HELP    HLP   62k   489  Dir  RO
PIP     COM     9k    68  Dir  RW          RENAME  COM    3k    23  Dir  RW
SET     COM    11k    81  Dir  RW          SETDEF  COM    4k    32  Dir  RW
SHOW    COM     9k    66  Dir  RW          SUBMIT  COM    8k    42  Dir  RW
TYPE    COM     3k    23  Dir  RW

Total Bytes     = 172k    Total Records =   1315    Files Found =   15
Total 1k Blocks = 172     Used/Max Dir Entries For Drive A:   19/  64
```

*read/write* što podrazumeva i pravo na brisanje) Ukoliko koristite CP/M PLUS možete da uradite i nešto mnogo bolje naredba poput SET * * [PASSWORD=RAČUNARI PRDTECT=DELETE] će onemogućiti bilo kom korisniku da obriše programe sa diskete osim u slučaju da zna lozinku RAČUNARI ! Premda dobro znamo da naši čitaoci *vrlo* rado slušaju o zaštiti programa ograničen prostor u ovom prikazu nam ne dozvoljava da se njom ozbiljnije bavimo Na ovu ćemo se temu svakako vratiti u nekom od naših budućih napisa

## Brisanje sa zadrškom

Pošto smo prilično dobro upoznali naredbu DIR upoznajmo i daleko opasniju naredbu ERA odnosno ERASE (na CP/M PLUS) Naredba ERA kao što pretpostavljate omogućava brisanje nepotrebnih datoteka sa diskete Zašto kažemo da je ova naredba opasna? Zato što bi nehotično ERASE * * obrisalo sve programe sa više disketa i tako eventualno izazvalo ogromnu štetu Zato je u CP/M i posebno u CP/M PLUS ugrađeno mnogo sigurnosnih brava ako računar proceni da brišete previše programa (otkucali ste na primer pomenuto ERA * * ) na ekranu će se pojaviti poruka poput ALL FILLES (Y/N)? ili ARE YOU SURE (Y/N)? Pritisnite slovo Y samo ako ste sigurni da znate šta radite! Čini nam se da je daleko bolje koristiti naredbu ERAQ *Erase With Query)* koja pre nego što obriše datotaku ispisuje njeno puno ime i zahteva od korisnika da pritisne slovo Y da bi odobrio njeno brisanje Opšti je savet da naredba ERA koristite isključivo ako se u imenu iza nje **ne nalazi ne jedan** džoker znak!

Za preimenovanje postojećih datoteka se koristi naredba REN (odnosno RENAME na CP/M PLUS) koju ćemo najbolje razumeti na primeru REN RAČUNAR COM = COMPUTER COM će datoteci čije je ime bilo COMPUTER COM dodeliti ime RAČUNAR CDM Važno je zapamtiti da naredba REName ne kopira fajl menja se jedino njegovo ime u direktorijumu Zbog toga bi naredba RENAME A PROG PRO = B PROG COM izazvala poruku o grešci jer smo pokušali da premestimo PROG COM sa drajva B na drajv A što se nikako ne bi moglo izvršiti bez kopiranja sadržaja CP/M će izdati poruku FILE EXISTS čak i ako napišete RENAME A=B pri čemu datoteka sa imenom A već postoji — kada bi neka druga datoteka zauzela njeno mesto operativni sistem ne bi znao šta da radi sa pokrivenom verzijom

Korisnici CP/M-a iako mnogo rade sa fajlovima retko upotrebljavaju naredbu SAVE koja omogućava snimanje podataka na disk Naredba SAVE je naime prilično ekscentrična umesto da počne sa snimanjem čim pritisnemo RETURN, računar će sačekati da pritisnemo CTRL C ili da se okonča izvršavanje tekućeg programa a onda će zahtevati ime (i tip) datoteka kao i adrese početka i kraja memorijske zone koja se snima (sve adrese se kucaju heksadekadno) Zbog svega toga se snimanje podataka na disk i njihovo učitavanju obično poverava eletemskim programima kao što je bejzik interpreter dok se SAVE komanda CP/M-a koristi samo kada treba na brzinu snimiti čitavu memoriju da bi se docnije posao mogao nastaviti na mestu prekida

## Multiprogramski rad

Naredba TYPE je na neki način suprotna od SAVE TYPE PROBA BAS će na ekranu prikazati sadržaj datoteke PRDBA BAS koja se pretpostavljamo sastoji samo od ASCII karaktera Pošto je broj linija na ekranu ograničen sadržaj iole veće datoteke bi bilo teško propratiti Zato se skrolovanje teksta zaustavlja pošto budu ispisane 24 linije dok za nastavak treba pritisnuti CTRL Q (ili, kod ranijih verzija CP/M-a bilo koji taster) Ukoliko vaš ekran ima više od 24 linije možete da otkucate TYPE PROBA-BAS P32 i tako saopštite računaru da se zaustavlja tek pošto ispiše 32 linije Ako vam ova konvencija ne odgovara otkucajte TYPE PROBA BAS NO PAGE pa ćete sami moći da zaustavljate skrolovanje sa CTRL S i da ga nastavljate sa CTRL D

Pomenimo i komandu USER koja premda je uključena u standard CP/M-a često ostaje neimplementirana CP/M je rekli smo namenjen radu jednog korisnika ali to ne znači da taj korisnik ne može da radi više stvari Ukoliko na iste diskete snimate programe koje razvijate tekstove koje pišete i baza podataka kojima povremeno pristupate, ubrzo ćete se naći u velikoj konfuziji Zato ćete pre nego što počnete sa programiranjem otkucati USER 1 pre nego što startujete tekst procesor USER 2 a pre nego što započnete rad sa bazom podataka — USER 3 Svaki od takvih fiktivnih korisnika se neće mešati u poslove drugih pa će vam tako komanda DIR davati samo podatka o datotekama koje ste kreirali dok ste imali odgovarajući korisnički broj Nema

| slika 5: | KONTROLNI KARAKTERI: |
|---|---|
| <CR> | Završava komandnu liniju. |
| <LF> | Prelazi u novi red. |
| CTRL J | Kao <LF>. |
| CTRL M | Kao <CR>. |
| CTRL U | Poništava komandnu liniju. |
| CTRL X | Kao CTRL U. |
| <BS> | Briše zadnji karakter. |
| <DEL> | Briše zadnji karakter i daje eho. |
| <RUB> | Kao <DEL>. |
| CTRL H | Kao <BS>. |
| CTRL K | Briše tekst do kraja linije. |
| CTRL E | Nastavak linije u drugom redu. |
| CTRL R | Ponavlja važeći deo linije. |
| CTRL S | Zaustavlja ispisivanje teksta. |
| STRL Q | Nastavlja ispisivanje teksta. |
| CTRL P | Uključuje i isključuje printer. |
| CTRL A | Kurzor desno. |
| CTRL B | Kurzor na početak (kraj) linije. |
| CTRL F | Ubaci SPACE na poziciju kurzora. |
| CTRL G | Kurzor levo. |
| CTRL C | Warm start CP/M-a |

čak nikakve prepreke da i kao USER 1 i kao USER 3 kreirate datoteku koja će se zvati PISMO TXT niti će nastupiti neka zabuna niti će jedan korisnik moći da promeni ili izbriše datoteku drugoga Broj korisnika kako kod CP/M-a tako i kod CP/M PLUS može da se nalazi između 0 i 15 pri čemu svi korisnici mogu da pristupaju datotekama korisnika 0 U korisničko područje 0 se dakle smeštaju sistemski programi i zajednička datoteka

U osnovne naredbe CP/M-a se ubrajaju i velika slova A B C itd iza kojih se stavlja dvotačka Napisavši B na primer stavljamo računaru do znanja da pri svim budućim SAVE ERASE i sličnim naredbama treba da pristupa drajvu B a ne početnom drajvu A Posle ova naredbe osnovni prompt će biti promenjen u B>

## Kontrolni karakteri

Već smo rekli da se listanja neke datoteka može privremeno zaustaviti sa CTRL S i da se nastavak tog listanja postiže sa CTRL Q S i i Q naravno, nisu jedini kontrolni karakteri koje CP/M koristi na slici 5 je prikazan sumarni pregled slova koja možete pritisnuti zajedno sa CTRL da biste pokrenuli ekranski editor obrisali deo linija startovali štampač ili izvršili neku drugu specijalnu funkciju Uz sav rizik da se većina čitalaca nasmeje reći ćemo da **nije potrebno** ciljati tastere CTRL i A kako bismo ih pritisnuli istovremeno Da otkucate CTRL A pritisnite dirku CTRL i, držeći je pritisnutu pritisnite ili otpustite taster A Tek posle toga možete da otpustite taster CTRL — računar je svakako već izvršio zadatu naredbu!

Skrećemo vam pažnju na komandu CTRL C koju treba koristiti posle bilo koje zamene disketa u nekom od drajvova (osim ako je tu zamenu zahtevao sam računar u kom se slučaju pretpostavlja da je programer znao šta radi) Sa CTRL C osim toga prekidamo rad svakog sistemskog programa kao što je bejzik interpreter i vraćamo kontrolu operativnom sistemu (prompt A> ) Iako moguć izlazak iz aplikativnih programa sa CTRL C nije preporučljiv jer može da ostavi neke datoteke otvorenima što obično čini da poslednja promene koje smo u njima izvršili ne budu uvažene Još je opasnije pritiskati dirku RESET koja je obično smeštena sa zadnje strane kompjutera ovaj je taster rezervisan samo za situacije u kojima nikako drukčije ne možemo da spasemo podatke! Zaključak koristite, kada je god to moguće naredbe poput QUIT da biste prekinuli rad nekog programa i vratili kontrolu CP/M-u

**Dejan Ristanović**

# Šta ima novo

## Bolji dani za 128

*Kada već spominjemo PC 128 i Komodor, treba reći da sa pojavom novog diska 1571 polako ali sigurno skače i prodaja ovog računara Za 128 se pojavilo preko 50 različitih knjiga a posebno je interesantan i impozantan broj CP/M programa Programi se pojavljuju u velikim količinama — već se mogu nabaviti svi bolji CP/M programi a kompatibilnost sa legendarnom šezdesetčetvorkom je dokazana 100% Znači hrpa igara i dosta odličnih poslovnih programa već u startu! Možda je to pravi izbor naravno u očekivanju evropske amige*

## Ada za „komodor"

*Za one vlasnike komodora koji ne žele da menjaju svoje ljubimce posle male pauze programi ponovo stižu u ogromnim količinama igre već postaju neverovatne i prelaze sve granice (Lukas film games — na primer) HACKER postaje legenda ali stižu i vrlo korisni i profesionalno urađeni korisnički programi Mnoge programere će sigurno interesovati da mogu da nabave i odličan C kompajler ali i adu koji su trenutni vodeći programski jezici Ne treba podsećati da je ada zvanični jezik Pentagona i da su na jeziku C pisani operativni sistemi za amigu i za atari 520*

## Sam svoj doktor

*Za komodor se nedavno pojavio program za lečenje u kući Program vam postavlja pitanja deli savete i na kraju postavlja dijagnozu Program zauzima dve diskete i napravljen je potpuno profesionalno Već posle prvih razgovora dobio sam savet da posetim lekara što je dokaz da program odlično radi*

## Amiga dolazi

*Amiga dolazi Polako ali sigurno pojedine firme i diileri već reklamiraju Commodora Amigu po obećanim cenama Očekujemo ozbiljnije vesti sa velikim interesovanjem*

## Hakeri igraju hakera

*Komodoristi — da li igrate HACKERA Nemoguće da još niste nabavili ovaj čuveni program Ako ga igrate pošaljite nam svoja iskustva i rešenja Još niko nije rešio HACKERA*

# Programiranje na mašincu

# sažimanje teksta

Često se za nekoga kaže On piše stenografskom brzinom Istina je stenografi pišu vrlo brzo ali koriste i neke „trikove" koji i nama mogu dobro doći Broj slova u tekstu značajno se smanjuje ako se iz njega izbace neki samoglasnici U većini slučajeva značenje reči se interpretira bez teškoća televizor — tlvzor Da značenje reči ipak ne bi došlo u sumnju slovo iza koga sledi neki samoglasnik piše se na poseban način — ili se spušta malo niže u odnosu na ostali tekst ili se nekako zaobli

### Interni kod

Slična tehnika se može primeniti i na računaru Pri upisivanju ASCII koda slova u memoriju setovanjem nekog bita moguće je naznačiti koje je slovo sledeće Sledeće slovo se ne memoriše posebno već poseban program čitajući bitove ASCII koda zabeleženih slova sam umeće znakove koji nedostaju Osim samoglasnika iz teksta se mogu odstraniti i grupe slova Stenografi takvu grupu nazivaju „sigla" U našem jeziku su česte sledeće grupe st,zd zn,sp pl spl pr tr kl kv,dv,vl Ispred ili iza sigle se pojavljuju samoglasnici Sem sigli u tekstu postoji i određen broj reči koji se ponavlje više puta Takvu reč ili siglu programeri nazivaju tokenom Token se u memoriju upisuje slovo po slovo samo jednom i to u posebnu tabelu Svaki token označen je svojim brojem koji ga u proizvoljnom tekstu zamenjuje Sve bejzik instrukcije i imena funkcija na spektrumu su tokenizovane tj u memoriji se predstavljaju samo jednim brojem Na primer naredba PRINT se beleži kao broj 245

Prema ASCII kodu sve znakove (slova brojevi specijalni znaci) moguće je predstaviti sa 7 bita Pošto u bajtu ima 8 bita poslednji bit možemo upotrebiti kako hoćemo Postoji međutim i mogućnost da se oslobodi još jedan bit Formiraćemo karakter set koji će se sastojati od velikih i malih slova brojeva i znakove ?' ! ,-' Ukupan broj znakova sada iznosi 63 Kako ASCII kodovi za navedene znake ne predstavljaju neprekidan niz brojeva potrebno je formirati interni kod od 0 do 62 To je učinjeno prema datoj tabeli

| Znak | ASCII kod | interni kod |
|---|---|---|
| a—z | 97—122 | 0—25 |
| A—Z | 65—90 | 26—51 |
| 0—9 | 48—57 | 52—61 |
| | 46 | 62 |

Slova Q I q zamenjena su blanko znakom i znakom pitanja Čini se zgodnim da se slova w x,y W,X,Y ekvivalentiraju sa

```
   CLEAR 29999
10 GO SUB 9000
11 REM ************************
12 REM  POCETAK GLAVNE PROGRAME
13 REM ************************
20 LET r=0:c=n
30 POKE 23659,9: INPUT LINE a$
35 IF a$="END" THEN SAVE "C.TXT"CODE n,65535-n+1: STOP
40 LET kr=LEN a$
50 LET i=1
60 IF i>kr THEN GO SUB 400: GO TO 20
70 LET c=CODE a$(i)
80 GO SUB 200
90 IF (i+1)<=kr THEN GO SUB 300
100 POKE n,c
110 LET n=n-1
120 LET i=i+1
130 GO TO 60
131 REM ************************
132 REM PODPROGRAM ZA KODIRANJE
133 REM    UNESENOG TEKSTA
134 REM ************************
201 IF c>143 THEN IF c<147 THEN LET c=c-112: RETURN
202 IF c>146 THEN IF c<150 THEN LET c=c-99: RETURN
203 IF c=32 THEN LET c=42: RETURN
204 IF c=63 THEN LET c=6 RETURN
205 IF c>96 THEN LET c=c-97: RETURN
212 IF c>64 THEN LET c=c 39: RETURN
220 IF  >47 THEN LET  =c+4: RETURN
230 LET c=62
240 RETURN
241 REM ************************
242 REM PODPROGRAM KOJI VRSI
243 REM  SAZIMANJE OKO ZNAKOVA:
244 REM               'a    e
245 REM LENTIRANJE ZNAKOVA:
247 REM ************************
300 LET  =CODE a$(i+1)
301 IF  =32 THEN LET  =c+64: LET i=i+1: RETURN
310 IF  =97 THEN LET  =c+128: LET i=i+1: RETURN
320 IF  =101 THEN LET  =c+19 LET i=i+1: RETURN
330 RETURN
400 REM ************************
401 REM ISPISIVANJE TEKUCIH PO-
402 REM DATAKA I SMESTANJE
403 REM      TOG TEKSTA
404 REM ************************
410 POKE n,255: LET  =
420 PRINT AT 1 15
430 LET u=kr  (r
440 PRINT AT 2  5
450 LET p=p+1
460 PRINT AT 3 15 p
470 LET kc!-0
480 IF  >  THEN RANDOMIZE U
5584: LET  =0: RETURN
4 0 IF k =32  THEN LET k
RANDOMIZE USR 5584
500 PRINT AT  kc a$
510 LET n=n+1: LET
520 GO TO 480
9000 REM ************************
9001 REM      INICIJALIZ
9002 REM ************************
9020 BORDER 0: PAPER 7:
     CODE  5 
9032 LET p=0: LET  =-170
65,96
9030 PLOT 0,172: DRAW 255
W 0, 64: DRAW -255,0: DRAW 0
9070 PRINT AT 1 1' Tekuca adi-s,
9040 PRINT AT 2,1' ustedalbajta
9045 PRINT AT 3 1' Poruka broj
9050 PRINT AT 5,7' PAPER 5: KOMP
RESOR TEKSTA
9060 PRINT AT 7,1,'A.Radovanovic
     986,'
9990 RETURN
9999 CLEAR ' SAVE "T,k.' LINE
 SAVE  CODE 'CODE 65197,170
```

prvih 6 UDG znakova Ovim se otvara mogućnost da korisnik definiše grafiku u okviru teksta ili slova ć č š ž Kôd slova se u memoriju upisuje kao bajt čiji je sadržaj prikazan na slici

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 |
|---|---|---|---|---|---|---|---|---|
| | slobodni bitovi | | interni kod znaka | | | | | |

### Sažimanje teksta

Kompresija teksta u našem programu zasniva se na upotrebi internog koda i slobodnih bitova U jedan bajt se upisuju dve informacije Šesti i sedmi bit kodiraju sledeće stanje 00—bajt sadrži samo znak predstavljen internim kodom 01—iza znaka upisanog u prvih 6 bita sledi blanko znak 10—iza znaka upisanog u prvih 6 bita sledi slovo a 11—iza znaka upisanog u prvih 6 bita sledi slovo e

Bilo da se pišu igre avanture ili obrazovni programi uvek je u datom trenutku potrebno na ekranu prikazati samo jedan deo teksta koji se naziva poruka Program dat na bejzik listingu prima poruke koje korisnik ukucava kodira ih izbacuje slova a i e kao i blanko znake i upisuje u memoriju Kraj poruke će biti označen bajtom čija je vrednost 255 U toku rada sa programom na raspolaganju je informacija o uštedi u memorijskom prostoru Program je izuzetno efikasan — prosečna ušteda je

**Svakom vlasniku „spektruma" je, bez sumnje, poznat program „Scrabble" — logička igra slaganja i ukrštanja reči. Uz ovaj program vezana je na prvi pogled neverovatna činjenica da računar operiše čak sa 11000 reči. Pretpostavljajući da svaka reč, u proseku, ima 4 slova, a svako slovo u memoriji računara zauzima 1 bajt, prostom računicom dolazimo do broja od 44000 bajtova. Međutim, „spektrum" raspolaže sa oko 42 K slobodnog RAM-a, u koji je, sem datoteke, bilo potrebno smestiti i program. Primenom posebnih tehnika može se obezbediti veoma gusto pakovanje teksta u memoriji i time značajno uštedeti na dragocenom memorijskom prostoru.**

```
00010   [illegible]
00020 ;------------------------
00030 [illegible]
00040 ;------------------------
00050 ;------------------------
00060 [illegible]
00070 [illegible]
00080 [illegible]
00090 [illegible]
00100 [illegible]
00110 [illegible]
00120 ;------------------------
00130 [illegible]
00140 [illegible]
00150 ;------------------------
00160 [illegible]
00170 [illegible]
00180 [illegible]
00190 [illegible]
00200       CAL [illegible]
00210       [illegible]
00220 ;------------------------
00230 ;HL [illegible]
00240 ; [illegible]
00250 ; [illegible]
00260 ;RCADR [illegible]
00270 ; [illegible]
00280 ; [illegible]
00290 ; u HL
00300 ;------------------------
00310 PRNT LD [illegible]
00320       CAL [illegible]
00330 ;------------------------
00340       POKE [illegible]
00350       [illegible]
00360 ;------------------------
00370 ; [illegible]
00380 ; [illegible]
00390 ; [illegible]
00400 ; [illegible]
00410 ; [illegible]
00420 ;------------------------
00430 PRNT  LD  A [illegible]
00440       CP  255
00450       RET Z
00460 ;------------------------
00470 ;R se stav [illegible]
00480 ; [illegible]
00490 ;desna [illegible] informac
00500 ; [illegible]
00510 ;se [illegible]
00520 ;RCII [illegible]
00530 ;[illegible] slova [illegible]
00540 ;[illegible] SCROLL
00550 ;------------------------
00560       PUSH AF
00570       AND 63
00580       CALL    KONV
00590       RST 16
00600       CALL    SCROL
00610 ;------------------------
00620 ;A se skida sa steka [illegible]
00630 ;si se [illegible]
00640 ; [illegible]
00650 ;ulevo [illegible]
00660 ;na pocetak bajta
00670 ;------------------------
00680       POP AF
00690       AND 192
00700       RLC A
00710       RLC A
00720 ;------------------------
00730 ;Ako je:
00740 ;A=0 [illegible]
00750 ; [illegible]
00760 ;A=1-[illegible]
00770 ; [illegible] blanko znaka
00780 ;A=2-[illegible] "a"
00790 ;A=3-[illegible] "e"
00800 ;------------------------
00810 ;A se [illegible] znaka [illegible]
00820 ;; [illegible]
00830 ;se njegovo stampanje uz
00840 ;poziv rutine za SCROLL
00850 ;------------------------
00860       JR  Z,DEKR
00870       DEC A
00880       JR  NZ,AA
00890       LD  A,32
00900       JR  STAMP
00910 AA    DEC A
00920       JR  NZ,EE
00930       LD  A,97
00940       JR  STAMP
00950 EE    LD  A,101
00960 STAMP RST 16
00970       CALL    SCROL
00980 ;------------------------
00990 ;Smanji se HL tako da uka
01000 ;zuje na adresu sledeceg
01010 ;karaktera. Skace se na
01020 ;pocetak petlje.
01030 ;------------------------
01040 DEKR  DEC HL
01050       JR  PRNT
01060 ;------------------------
01070 ;Rut na koja [illegible]
01080 ;[illegible] ekrana za 1 red kada
01090 ;se ispise 32 karaktera
01100 ;
01110 ;D [illegible]
01120 ;se [illegible]
01130 ;D(". [illegible]
01140 ;-----
01150 SCROL INC D
01160       LD  A,D
01170       CP  32
01180       RET NZ
01190 ;------------------------
01200 ;Ukoliko je D=[illegible]
01210 ;se vraca na pozi[illegible] 10
01220 ;0 (pozicija je [illegible])
01230 ;------------------------
01240 SCRL  LD  A,[illegible]
01250       RST 16
01260       LD  A,10
01270       RST 16
01280       XOR A
01290       RST 16
01300 ;------------------------
01310 ;HL se stavi na stek a za
01320 ;tim se poziva SCROLL ru
01330 ;tina iz ROM-a Na ovo me
01340 ;sto mozete umetnuti svo
01350 ;ju rutinu za SCROLL
01360 ;HL se zatim sk[illegible]
01370 ;ka; D se postavi na
01380 ;0(pocetak novog [illegible])
01390 ;------------------------
01400       PUSH    HL
01410       CALL    3537
01420       POP HL
01430       LD  D,0
01440       RET
01450 ;------------------------
01460 ;Rutina koja na osnovu
01470 ;broja poruke u BC [illegible] adre
01480 ;se pocetka datoteke u HL
01490 ;racuna adresu na kojo
01500 ;je vasa poruka. Izracuna
01510 ;ta adresa se smesta u HL
01520 ;------------------------
01530 ;Izracunavanje se vrs[illegible]
01540 ;pomocu dve While petlje
01550 ;
01560 ;While BC<>0 do
01570 ; While (HL)<>255 do
01580 ;  dec HL end While
01590 ; dec BC
01600 ;end While
01610 ;------------------------
01620 RCADR LD  A,B
01630       OR  C
01640       RET Z
01650       LD  [illegible]
01660 WO1   CP  [illegible]
01670       JR  [illegible]
01680       DEC [illegible]
01690       JR  WO1
01700 WO2   DEC HL
01710       DEC BC
01720       JR  RCADR
01730 ;------------------------
01740 ;Proces vra[illegible]
01750 ;kompresovanog [illegible]
01760 ;va u ASCII kod [illegible]
01770 ; sa predstavlja sadr[illegible]
01780 ;akumulatora
01790 ;------------------------
01800 ;Izracunavanje se obavlja
01810 ;po obrnutom postupku k
01820 ;oji je koriscen u BASIC
01830 ;------------------------
01840 KONV  CP  22
01850       JR  C,UDOI
01860       CP  25
01870       JR  NC,UDOI
01880       ADD A,122
01890       RET
01900 UDOI  CP  48
01910       JR  C,POCK
01920       CP  51
01930       JR  NC,POCK
01940       ADD A,95
01950       RET
01960 POCK  CP  62
01970       JR  NZ,PRVI
01980       LD  A,46
01990       RET
02000 PRVI  CP  42
02010       JR  NZ,PRV2
02020       LD  A,32
02030       RET
02040 PRV2  CP  16
02050       JR  NZ,PRV3
02060       LD  A,63
02070       RET
02080 PRV3  CP  52
02090       JR  C,PRVA
02100       SUB 4
02110       RET
02120 PRVA  CP  26
02130       JR  C,KRP
02140       ADD A,39
02150       RET
02160 KRP   ADD A,97
02170       RET
02180 ;------------------------
02190 ;Adresa pocetka datoteke
02200 ;kompresovanog teksta
02210 ;------------------------
02220 BAZA  DEFW    65196
```

od 25% do 30% Na svakih 1000 znakova uštedi se od 250 do 300 bajtova Postepenim popunjavanjem memorije dolazi se do iznenađujuće činjenice da je u računar stalo više podataka nego što ih RAM fizički može da primi

*Unošenje programa*

1) Učitati neki program asembler (Gens Zeus) ostavljajući slobodne memorijske lokacije iznad adrese 65190

2) Ukucati bejzik listing

3) Startovati asembler ukucati priloženi asemblerski listing i prevesti ga na mašinski jezik

4) Vratiti se u bejzik i sa GOTO 9999 snimiti kôd na traku

5) Program će se snimiti iz dva dela koja predstavljaju program „Kompresor

*Tok rada sa kompresorom*

1) Sa LOAD učitati snimljeni program

2) Na papiru pripramiti tekst razdeljen na poruke numerisane od 0 naviše

3) U računar unositi jednu po jednu poruku Kraj rada označava se porukom END Ona snima na traku sadržaj memorije počevši od adrese koja stoji iza natpisa Tekuća adresa To je ujedno i najviša adresa do koje program koji koristi kompresovani tekst sme ići (To važi samo za mašinske programe i POKE instrukciju)

*Uključivanje kompresovanog teksta u program*

1) Naredbom CLEAR postaviti RAMTOP ispod adresa koju je program za kompresiju ispisao neposredno pre snimanja

2) Sa LOAD " CODE učitati kompresovani tekst sa trake

3) Pišite program tako da poruke pozivate po njihovim rednim brojevima To se vrši na sledeći način

LET por=34218 GOSUB 9997
9997 POKE 23729 (INT(por/256))
9998 POKE 23728 (por− INT(por/256))
9999 RANDOMIZE USR 65197

4) Broj poruke može biti u opsegu 0—65535

*Za one koji eksperimentišu*

Program za dekompresiju uzima broj poruke iz sistemske promenljive 23728 Ukoliko se program poziva iz mašinskog jezika broj poruke se stavlja u BC registarski par, a iz listinga se izbacuje instrukcija LD BC (2372) Program za dekompresiju automatski vrši SCROLL celog ekrana posle svakog ispisanog reda instrukcija CALL 3582 može se zameniti nekom SCROLL rutinom Pri tome treba obratiti pažnju da broj iza ORG treba smanjiti za onoliko koliko je dugačka ta rutina Broj iza labele BAZA i promenljivu n u bajziku inicijalizovati na vrednost koja je za jedan manje od ORG adrese Korišćenjem rutina za kompresiju slika i tekstova mogu se pisati bogato ilustrovane i sadržajne igre avantura

*A.R.*

*Kako se koristi*

# devpac mons

Ni jedan iskusan programer ne očekuje da će mu program proraditi od prve. Jednostavno bez i jedne trunke emocija treba prihvatiti činjenicu da izvršenje našeg najnovijeg programa protiče na način koji nismo mogli ni sanjati. Dešava se to svakoma pri startovanju bilo kakvog programa i pri izboru bilo kog programskog jezika.

Sa bejzikom je stvar nešto prostije jer bar ne može nikad (gotovo nikad) doći do blokiranja računara. Čim vidimo da program radi nešto što mu nije posao pritisnućemo BREAK i potražiti grešku u listingu.

Mašinski programi međutim prireduju daleko veće teškoće. Čak i kada se program izvrši do kraja ne možemo baš biti sigurni da li je on uredio samo ono što smo želeli ili je možda usput izveo i nešto po svom nahođenju.

U takvim situacijama kada nismo sigurni šta program zapravo radi ili kada nam nije jasno zašto uopšte neće da radi (a trebalo bi da radi) pozvaćemo u pomoć *disassembler* — program koji olakšava testiranja mašinskih rutina i obavlja niz drugih korisnih poslova. Pomoću disassemblera možemo jednostavno menjati naredbe mašinskog programa, izvršavati program korak po korak, u bilo kom trenutku pročitati sadržaj procesorskih registara ili sadržaj bilo kog bloka memorije. Jednostavno rečeno bez disassemblera se ne može ni zamisliti razvoj bilo-kakve komplikovanije mašinske rutine.

## *Šta je MONS*

Disassembler pod nazivom MONS nabavlja se paketu sa odgovarajućim asemblerom. Proizvođač je softverska firma Hisoft, a čitav paket je poznat pod imenom devpac. Namenjen je računarima sa procesorom Z80, a mi ćemo konkretno ovde govoriti o verziji za spektrum sa oznakom MONS3M. Razlika među pojedinim verzijama su minimalne i na njima se nećemo zadržavati.

Program MONS3M zauzima tačno 5800 bajtova i može se startovati sa bilo koje adrese u RAM-u. Najpodesnije je smestiti ga u sam vrh memorije, recimo na 58000

*LOAD CODE 58000*

Prvo startovanje programa vrši se od početne adrese

*RANDOMIZE USR 58000*

Međutim svako sledeće startovanje vrši se od adrese koja je za 29 veća od početne

*RANDOMIZE USR 58029*

Ove promene početne adrese uslovljena je izmenama u samom MONS-u u toku njegovog prilagođavanja adresi na koju je učitan.

Neposredno po startovanju programa nakon kraće pauze i poruke proizvođača ugledaćemo ekran popunjen slovima i ciframa, gotovo rekli bi se sračunat tako da

```
C372  C9        RET
 PC C372   C9 00 00 00 00 00 00
 SP 7518   2B 2D 65 33 00 40 ED
 IY 5C3A   FF ED 00 2C 75 2C 75
 IX 03D4   04 0C 0D 20 FD 0E 3F
 HL 5CDD   00 50 02 00 EA 0D 00
>DE 5D12   00 90 E2 00 00 0A 00
 BC 0000   F3 AF 11 FF FF C3 C8
 AF E720
 IR 3F45

C366  CC    C36E  19    C376  00
C367  88    C36F  C4    C377  00
C368  19    C370  E5    C378  00
C369  E1    C371  19    C379  00
C36A  E8   >C372  C9<   C37A  00
C36B  A7    C373  00    C37B  00
C36C  BD    C374  00    C37C  00
C36D  52    C375  00    C37D  00
```

prestraši početnike. Međutim radi se o vrlo dobro organizovanoj i preglednoj tabeli, iz koje možemo saznati sve što bi nas interesovalo u toku disassembliranja.

Englezi su ovaj ekran označili imenom „Front panel display", a mi možemo usvojiti naziv „radna tabela". Prvo ćemo svakako objasniti šta znači svaki znak u toj tabeli.

## *Radna tabela*

Primer jedne radne tabele dat je na našoj slici. U donjoj polovini ekrana uvek se nalazi prikazan blok od 24 bajta memorije u okolini adrese koju unapred izaberemo. U našem primeru izabrana je adresa &C372 i njen sadržaj je kao što se vidi &C9. Znaci > i < okružuju zadatu adresu i predstavljaju tzv. *pokazivač memorije* („memory pointer").

U samom vrhu ekrana se nalazi disassemblirana mašinska naredba adresirana memorijskim pokazivačem. U našem primeru radi se o naredbi **RET**.

Zatim sleda sadržaji svih registara procesora Z80. Znak > je *pokazivač registra* i služi nam da izaberemo registar koji želimo da izmenimo. Sadržaj svakog registra je ispisan u istom redu desno u heksadekadnom obliku. Recimo HL sadrži &5CDD, a akumulator A sadrži &E7. Registar F je trenutno postavljen na &20, što znači da ni jedan indikator nije setovan (setovan je samo bit 5, ali kao što znamo bitovi 3 i 5 registra F se ne koriste). Kada je neki od indikatora setovan onda se desno od sadržaja registra F ispisuje i oznaka odgovarajućeg indikatora. Recimo, u slučaju F=&40 setovan je indikator nule „Z" pa bismo na ekranu ugledali

AF &E740 Z

Svaki indikator ima svoje slovo, pa ako je to slovo ispisano znači da je indikator setovan. U suprotnom je resetovan.

Kada su svi indikatori setovani imamo ispis

SZ H VNC

Prazna polja odgovaraju nekorišćenim bitovima 3 i 5. Slova se odnose na sledeće indikatore:

S — *indikator znaka* ( sign flag )

Z — *indikator nule* ( zero flag )

H — indikator poluprenosa ( half carry flag )

V — *indikator parnosti/prekoračenja* ( parity/overflow flag )

N — *indikator oduzimanja* ( subtract flag )

C — *indikator prenosa* ( carry flag )

Što se tiče 16-bitnih registara PC, SP, IY, IX, HL, DE, BC desno od njihovog sadržaja nalazi se niz od po sedam bajtova počev od adrese koju registar trenutno sadrži. Tako na primer vidimo da na adresi &5CDD (sadržaj registra HL) počinje blok bajtova &00 &50 &02 &00 itd.

U samom dnu ekrana nalazi se kursor spreman za prihvatanje naših naredbi. Prelaz sa malih na velika slova i obrnuto (kursori L i C) vrši se na uobičajen način (CAPS LOCK).

## *Naredbe*

MONS3M dopušta veliki broj naredbi koje korisnik unosi sa tastature. Svaka naredba je u stvari *jedno ukucano slovo* bez potrebe za tasterom ENTER. Izvršenje naredbe počinje čim je ukucamo, što ponekad može da nam pričini neprijatnosti ukoliko pritisnemo taster koji nismo želeli. Zato pri radu sa MONS-om moramo biti vrlo obazrivi i dobro promisliti pre svakog poteza.

Pogrešne (nepostojeće) naredbe biće jednostavno ignorisane. Radna tabela se menja posle svake izvršene naredbe tako da uvek možemo videti rezultate onoga što smo upravo izveli.

**1. EDIT (CAPS SHIFT 1)**

Naredba **EDIT** obavlja povratak u bejzik, tačnije skok u glavnu izvršnu petlju bejzik interpretera. Ponovo pozivanje MONS-a vrši se od adrese za 29 veće od početne.

**2. # (SYMBOL SHIFT 3, povisilice)**

Ovom naredbom zahteva se ispisivanje svih adresa u radnoj tabeli u *decimalnom* obliku umesto uobičajene heksadekadne forme. Istom naredbom vrši se i vraćanje u heksadekadni ispis.

**3. ENTER**

Pomeranje pokazivača memorije za jedan bajt unapred.

**4. ↑ (CAPS SHIFT 7, strelica gore)**

Pomeranje pokazivača memorije za jedan bajt unazad.

**5 → (CAPS SHIFT 8, strelica desno)**

Pomeranje pokazivača memorije za osam bajtova unapred.

**6. ← (CAPS SHIFT 5, strelica levo)**

Pomeranje pokazivača memorije za osam bajtova unazad.

**7 Izmena sadržaja memorije**

Kada želimo da promenimo sadržaj bajta adresiranog pokazivačem memorije ukucaćemo novu heksadekadnu vrednost i zatim pritisnuti **ENTER**, ili bilo koji drugi komandu za pomeranje pokazivača memo-

***Nabavka jednog dobrog komercijalnog programa, u našim uslovima, predstavlja pravu malu avanturu. Najčešće nemamo dovoljno novca da bismo kupili originalnu verziju programa u Engleskoj, pa smo, tako, prinuđeni da se okrenemo našem razvijenom piratskom tržištu. Naše softverska „firma", međutim, ne vole rad sa papirom i „svoje" programe isporučuju obično bez uputstava. Ako uputstvo i stigne, brzo shvatamo da od njega nema mnogo koristi — uputstvo je pisano na engleskom, a mi smo u školi učili ruski. Sledi, dakle, potraga za prevodom . . . Redakcija „Računara" je, da bi pomogla svojim čitaocima, odlučila da prikaže kvalitetnije komercijalna programe, sa uputstvom za njihovo korišćenje.***

### Program. DELETE

#### Test primer za rad sa MONS-om

```
L_ADOR  EQU   &196E       Rutina LINE_ADDR iz ROM-a
NXTONE  EQU   &19B8       Rutina NEXT_ONE iz ROM a
RCLM 1  EQU   &19E5       Rutina RECLAIM_1 iz ROM-a
REP_D   EQU   &288B       Izdavanje raporta Q
E_PPC   EQU   &5C49       Sistemska promenljiva
        ORG   50000
DELETE  LD    (E_PPC) DE  Postavljanje programskog kursora
        AND   A           Brisanje indikatora prenosa
        SBC   HL DE       Da li je početak veći od kraja?
        JP    C REP_Q     Ako jeste prijavi grešku
        ADD   HL DE       Obnovi konačnu liniju
        EX    DE HL       Početak u HL kraj u DE
        PUSH  DE          Sačuvaj krajnju liniju
        CALL  L_ADDR      Nađi početnu adresu
        EX    (SP) HL     Početna adresa na stek krajnja linija u HL
        CALL  L_ADDR      Nađi adresu krajnje linije
        LD    E L         Kopiraj tu adresu u
        LD    D H         registar DE
        CALL  Z NXTONE    Ako je linije nađena pomeri se iza nje
        POP   HL          Početna adresa u HL
        EX    DE HL       Početna adresa u DE krajnje u HL
        AND   A           Briši indikator prenosa
        SBC   HL DE       Da li su početak i kraj isti?
        ADD   HL DE       Obnovi kraj
        CALL  NZ RCLM_1   Izbaci blok samo ako početak i kraj
                          nisu isti
        RET               Povratak u bejzik
```

## Monsom po programu

*Ilustrovaćemo rad sa MONS-om na primeru jednog jednostavnog programa čiji listing objavljujemo Radi se o operaciji DELETE koja izbacuje blok bejzik linija iz programa Na ulazu DE sadrži linijski broj početne a HL linijski broj krajnje linije Već je prethodno provereno da oba linijska broja leže u korektnom opsegu (1 — 9999)*

*Programski kursor se postavlja u liniju DE*

*Ukoliko je konačni linijski broj manji od početnog program treba da prijavi grešku O Parameter error U protivnom mora iz bejzika izbaciti sve linije počev od DE (ili od prve sledeće ako takva ne postoji) pa zaključno sa HL (ili prvom prethodnom ako takva ne postoji) Ako je HL jednako DE izbacuje se samo jedna linija ali ako takve nema ništa se neće preduzeti*

*Resetovaćemo računar i izvršiti **CLEAR 29999** Zatim učitavamo GENS na 30000 i MONS na 58000 Startujemo GENS i ukucamo ceo program DELETE naravno bez komentara Asembliramo ga i vratimo se u bejzik*

*U bejziku ukucamo kratak program nevažnog sadržaja*

*10 REM*
*20 REM*
*30 REM*
*40 REM*
*50 REM*
*60 REM*
*70 REM*
*80 REM*
*90 REM*

*Konačno startujemo MONS na adresi 58000*

*Potražićemo pre svega naš program u memoriji Početak mu je na 50000 pa zato izvršimo **N 50000**. Kao odgovor ćemo dobiti heksadekadnu adresu &C350 Zato dalje možemo uraditi **M C350** čime dovodimo pokazivač memorije na prvi bajt programa Još moramo pripremiti i registar PC Ukucavamo C350 i pritiskamo tačku Time je sve spremno za izvršenje programa korak po korak Jedino nismo pripremili registre DE i HL kao ulazne parametre*

*Recimo da želimo izbacivanje linija 40 — 70 Tada u HL treba staviti &0046 (tri puta tačka 0046 tačka) a u DE &0028 (tačka 0028 tačka)*

*Možemo malo zastati i pogledati izgled redne tabele U vrhu ekrana je prva naredba programa **LD (E_PPC) DE** Registar IY sadrži &5C3A što je neophodno ako zovemo potprograme iz ROM-a Registri PC HL i DE su spremni za izvršenje našeg programa a SP sadrži adresu &7518 Primećujemo da je poslednja adresa na steku &2D2B inače uobičajena za povratak u bejzik (STACK_BC)*

*Mirno možemo izvršiti prvu naredbu pritiskom na  Ako nas interesuje da li je naredba zaista izvršena pročitaćemo sadržaj E_PPC jednostavnom promenom pokazivača memorije **M 5C49** Videćemo da je vrednost DE registra stvarno tamo Zato se vraćamo nazad na adresu sadržanu u PC (sledeća naredba našeg programa) **M C354***

*Izvršimo bez mnogo razmišljanja i sledeće dve naredbe **AND A i SBC HL DE***

*Sledi veoma opasna naredba **JP C, REP_Q** Da je nekim slučajem u ovom trenutku setovan indikator prenosa a mi izvršimo komandu  došlo bi do blokade izveštaji za grešku nisu mogući iz MONS-a Međutim pošto je kod nas C =0 slobodno možemo izvršiti i tu naredbu*

*Sledeće tri naredbe su takođe sasvim bezopasne Posle **PUSN DE** možemo samo obratiti pažnju na novo stanje mašinskog steka Videćemo da je SP umanjen za dva kao što i treba očekivati*

*Nova uzbuna nastaje na naredbi **CALL L_ADDR** Pozivanje potprograma iz ROM-a nije uvek preporučljivo ako ne znamo tačno šta će se desiti Međutim ovde se radi o programu koji doslovno za svaki ulazni podatak obavlja uspešan povratak A osim toga naš ulazni podatak sigurno ima smisla (linija 40 postoji u programu bez sumnje) Dakle slobodno možemo izvršiti **CALL L_ADDR** koristeći naredbu > Nema razloga da tamo idemo korak po korak ali vama savetujemo da to probate kao vežbu*

*Na povratku iz L_ADDR HL će biti postavljen na početak linije 40 (adresa &5CDD) a DE je na početku prethodne indikator nule je setovan jer je pronađena linija koja je i tražena*

*Posle naredbe **EX (SP), HL** ponovo imamo **CALL L_ADDR.** za šta opet koristimo komandu > Slede dve jednostavne naredbe a onda opet oprez **CALL Z NXTONE** Pošto je indikator nule setovan do poziva će sigurno doći Jedino se treba odlučiti da li pozvati potprogram sa  ili ga izvršavati korak po korak Taj potprogram baš nije blagonaklon prema bilo kakvim ulaznim podacima ali kod nas je sve u redu i slobodno možemo izvršiti > Ostatak programa nije nikakav problem sve do poslednje naredbe koja poziva rutinu RECLAIM iz ROM-a a tu se kriju mnoge zamke Na ulazu nipošto ne smeju HL i DE biti jednaki Naš program o tome sam vodi računa Dalje DE mora biti na prvom a HL na poslednjem bajtu koji se izbacuje Kod nas je očigledno sve u redu i mirno možemo izvršiti >*

*Tako smo bez većih teškoća stigli do naredbe **RET** Redna tabela ima izgled upravo kao na slici koju objavljujemo (jedino registar R može imati drugačiju vrednost) Vratićemo se u bejzik naredbom **EDIT** i uveriti se da u programu nedostaju linije 40 50 60 i 70*

rija (naredbe pod brojevima 3, 4, 5 i 6). Vrednost zadatog bajta biće nakon toga izmenjena, a pokazivač memorije postavljen na novu adresu.

**8. M**

Naredbom **M** može se postaviti pokazivač memorije na bilo koju zadatu adresu ukucavanjem odgovarajuće heksadekadne vrednosti iza **M**. Tasterom **ENTER** završavamo ukucavanje adrese.

**9. • (SYMBOL SHIFT M, tačka)**

Naredba • pomera pokazivač registra za jedno mesto unapred. Kretanje je ciklično u redosledu PC IY IX HL DE BC AF. Recimo, ako je pokazivač trenutno na PC a mi želimo da ga dovedemo na HL, moramo tri puta izvršiti •. Vraćanje na PC zahteva još četiri iste operacije. Registri SP i IR se ne mogu adresirati.

**10. Promena sadržaja registra**

Sadržaj registra na kome upravo stoji pokazivač može se izmeniti direktno ukucavanjem novog heksadekadnog sadržaja praćenog tačkom. Pri tome se položaj pokazivača neće promeniti.

**11. G**

Naredba **G** vrši pretraživanje memorije počev od trenutnog stanja pokazivača memorije, tražeći zadati niz bajtova. Čim pritisnemo **G**, od nas se očekuje da unesemo niz bajtova za kojim će se tragati. Ukucavamo redom bajt po bajt u heksadekadnom obliku, a odvajamo ih međusobno sa **ENTER**. Kada je niz kompletiran, još jednom pritiskamo **ENTER**, i traganje počinje. Zadati niz će *uvek* biti pronađen, ako ne na drugom mestu, onda u samom MONS-u, tamo gde smo ga sami ukucali. Čim se niz otkrije, pokazivač memorije se postavlja na njegov početak i vraće se u radnu tabelu.

**12. N**

Naredba **N** pronalazi u memoriji *sledeće* pojavljivanje niza za kojim je tragala prethodna naredba **G**. Na taj način, koristeći naredbu **G** i zatim uzastopno **N**, moguće je pronaći u memoriji sva mesta gde postoji zadati niz.

**13. I**

Naredbom I vrši se kopiranje bloka bajtova sa jednog mesta na drugo. Uslediće pitanja *First* i *Last* na koja moramo odgovoriti početnom i krajnjom heksadekadnom adresom bloka koji kopiramo, dok iza *To* ukucavamo adresu na koju blok šaljemo. To može biti čak i neka adresa unutar samog bloka koji se kopira. Slovo *I* se u stvari odnosi na "inteligentno" kopiranje.

**14. H**

Pretvaranje dekadnog broja u heksadekadni. Iza **H** treba ukucati dekadni broj koji prevodimo, praćen sa **ENTER**. Rezultujući heksadekadni niz će se pojaviti u produžetku istog reda.

**15. Q**

Prelazak sa glavnog seta registra (HL DE BC AF) na alternativni (HL' DE' BC' AF') i obrnuto.

**16. P**

Popunjavanje izabranog bloka memorije zadatim bajtom. Na pitanja *First* i *Last* odgovaramo početnom i krajnjom adresom bloka, a iza *With* zadajemo bajt kojim treba popuniti blok.

**17. L**

Listanje bloka bajtova počev od adrese pokazivača memorije. Listanje se vrši se po četiri bajta u jednom redu, praćeno odgovarajućom ASCII interpretacijom. Pri tome se bajtovima većim od &7F oduzima &80 (resetuje se bit 7). Bajtovi manji od &20 prikazuju se tačkom.

Svaku novu stranicu listinga dobijamo pritiskom na bilo koji taster osim **EDIT** (povratak u bejzik) i ← (povratak u radnu tabelu).

**18. " (SYMBOL SHIFT P, znak navoda)**

Ova naredba obavlja isti posao kao i **L**, s tim što se izlaz vrši na štampač.

**19. Y**

Naredbom Y može se u memoriju počev od trenutnog položaja pokazivača memorije ubaciti proizvoljan ASCII niz bez potrebe za ukucavanjem heksadekadnih vrednosti znakova. U dnu ekrana će se pojaviti nova linija u koju unosimo niz slovo po slovo. Kao oznaku za kraj treba ukucati ←. Nova vrednost pokazivača memorije biće na prvoj adresi iza unetog niza.

**20. $ (SYMBOL SHIFT 4, dolar)**

Naredba $ obavlja disasembliranje bajtova počev od trenutnog položaja pokazivača memorije. Listing se prikazuje na ekranu i može se produžavati pritiskom na bilo koji taster osim **EDIT** (povratak u bejzik). Vraćanje u radnu tabelu obavlja se takođe sa $.

**21. T**

Naredba **T** je daleko moćnija od $. Moguće je disasemblirati proizvoljan blok bajtova i formirati tekst koji će kasnije biti prihvaćen od strane GENS asemblera.

Na pitanja *First* i *Last* i ovde odgovaramo početnom i krajnjom adresom bloka. Ako krajnja adresa bude manja, izvršenje naredbe će biti obustavljeno.

Na pitanje *Printer?* odgovorićemo sa *Y* ako želimo listanje na štampaču. Bilo koji drugi odgovor (najjednostavnije **ENTER**), podrazumeva samo listanje na ekranu.

Pitanje *Text* zahteva od nas adresu u memoriji na kojoj će se formirati tekst disasembliranog programa. Ako ne želimo da se tekst formira, pritisnućemo samo **ENTER**.

Ukoliko smo zahtevali generisanje teksta, uslediće pitanje *Workspace*. Moramo ukucati adresu na kojoj će disasembler formirati tabelu sa simboličkim adresama. Za svaku simboličku adresu biće utrošena dva bajta radnog prostora. Ako umesto odgovora ukucamo samo **ENTER**, podrazumeva se radni prostor na &6000.

Konačno, od nas će se tražiti da definišemo sve blokove podataka koji eventualno postoje u programu i koje stoga treba isključiti iz procesa disasembliranja uz jednostavno generisanje **DEFB** naredbi. Uslediće niz pitanja *First* i *Last* na koja odgovaramo početnom i krajnjom adresom bloka podataka. Sve te adrese MONS će pamtiti u prostoru iza sopstvenog koda. Prema tome, broj blokova podataka koje možemo zadati ograničen je jedino raspoloživim memorijskim prostorom iza MONS-a. Svaki blok zahteva četiri bajta.

Kad budemo zadali sve blokove podataka, na poslednji par pitanja *First* i *Last* treba odgovoriti sa **ENTER**.

Ekran se briše i počinje disasembliranje uz listanje odgovarajućeg teksta. Privremeni prekid može se ostvariti pritiskom na **ENTER** ili **SPACE**. Nastavak se tada vrši bilo kojim tasterom, dok ← obavlja povratak u radnu tabelu.

Na kraju disasembliranja, ako smo zahtevali formiranje teksta, imaćemo poruku *End of text* iza koje stoji adresa prvog bajta iza samog teksta. Razlika te adrese i početne daće dužinu teksta. Možemo se vratiti u bejzik i snimiti generisani tekst na kasetu da bismo ga kasnije uzeli u asembler naredbom **G**.

Međutim, ako se GENS istovremeno nalazi u memoriji, tekst možemo formirati odmah tamo gde ga asembler i očekuje. Procedura u tom slučaju bila bi sledeća.

Prvo u GENS-u izvršimo naredbu **X**. Od dve adrese koje nam pri tom budu ponuđene uzećemo prvu u slučaju da želimo brisanje već postojećeg programa, ili drugu ako želimo da se nadovežemo na postojeći program. Naravno, ako nikakvog programa u GENS-u nema, obe adrese su iste i uzećemo bilo koju. Vraćamo se u MONS i tu adresu navodimo u okviru naredbe **T**, kao odgovor na pitanje *Text*. A kada na kraju dobijemo adresu završetka teksta, moramo je upisati u GENS-ovu sistemsku varijablu *TEXTEND* (to su dva bajta koja počinju na adresi za 54 većoj od početne adrese GENS-a). Jednostavno ćemo iz MONS-a izvršiti naredbu **M** i postaviti pokazivač memorije na *TEXTEND*, a onda u te dva bajta upisati zaključnu adresu teksta (prvo bajt manje težine).

Vratićemo se zatim u bejzik i startovati GENS (vruć start). Primetićemo da tekst koji je formirao MONS nije baš najpregledniji, jer nedostaju kodovi za tabulaciju (tekst je maksimalno kondenzovan). To se može ispraviti naredbom **c** u okviru GENS-a.

Sve simboličke adrese koje MONS generiše počinju slovom L ispred apsolutne heksadekadne adrese (npr. L75CD).

**22. (SYMBOL SHIFT Z, dve tačke)**

Naredba koristi se pri izvršavanju mašinskog programa korak po korak (ako izvršavamo program iz ROM-a, to je i jedini način). Izvršiće se naredba koju trenutno adresira pokazivač memorije, ali i registar PC *mora obavezno sadržati istu adresu*. U protivnom će doći do kraha.

**23. W**

Postavljanje *prekidne tačke* na adresu određenu pokazivačem memorije. Prekidna tačka služi MONS-u kao znak da na tom mestu treba prekinuti izvršenje programa (ukoliko je izvršenje u toku) i obaviti povratak u radnu tabelu. Primetićemo da po izvršenju naredbe **W**, tri bajta našeg programa bivaju zamenjena jednom **CALL** instrukcijom koja označava prekid. Naša tri bajta se čuvaju i biće obnovljena čim prekidna tačka bude obavila svoju funkciju. Pored te tri bajta mora se čuvati i adresa prekidne tačke, što sve skupa znači pet bajtova potrebnog memorijskog prostora po svakoj prekidnoj tački. Njih može u programu biti proizvoljan broj, sve dok ima dovoljno memorije iza MONS-a (naredba **W** koristi isti onaj prostor kao i naredba **T**, o čemu moramo voditi računa, jer će svaki poziv naredbe **T** upropastiti naše prekidne tačke).

**24. + (SYMBOL SHIFT K, plus)**

Nastavak izvršenja programa počev od trenutnog sadržaja programskog brojača PC (stanje pokazivača memorije nije pri

toma bitno) Program se zaustavija kad nađe na prekidnu tačku uz povratak u radnu tabelu

**25 > (SYMBOL SHIFT T, veće)**

Postavljanje prekidne tačke iza tekuće naredbe i nastavak izvršenja programa Jednostavnije rečeno biće izvršena naredbe adresirana pokazivačem memorije i registrom PC na potpuno isti način kao da smo sami postavili prekidnu tačku iza te naredbe i onda izvršili komandu + Očigledna je razlika ova naredba od Naredbom ' ća recimo, u slučaju **CALL** naredbe samo biti obavljen skok na prvu naredbu potprograma koji se poziva dok će naredba > izvršiti čitav potprogram i vratiti se na naredbu iza **CALL** To je korisno kada smo sigurni da potprogram radi bez problema tako da nema potrebe da se i po njemu šetamo korak po korak

**26 J**

Izvršenje mašinskog programa počev od zadate adrese Pre nego što ukucamo samu heksadekadnu adresu možemo obustaviti ovu naredbu komandom U protivnom skok se više ne može sprečiti Povratak u radnu tabelu usledića jedino pri nailasku na prekidnu tačku

Naredba **J** je zgodna kada želimo da se vratimo u bejzik a bojimo se da to baš neće biti uspešno posle nekih zahvata koje smo iz MONS-a izveli u prostoru sistemskih varijabli i mašinskog steka Tada prosto treba izvršiti **J &11B7**, što je skok na naredbu **NEW** Razume se RAMTOP pri tome treba da je ispod MONS-a jer će u protivnom i MONS biti uništen Što se tiče bejzik prostora on se ovim nikadko ne može spasiti

**27. , (SYMBOL SHIFT N, zarez)**

Postavljanje pokazivača memorije na adresu sadržanu trenutno u vrhu mašinskog steka Vrh mašinskog steka s druge strane određen je registrom SP

**28 O**

Naredba O postavlja pokazivač memorije na novu vrednost određenu relativnim skokom **JR ili DJNZ** naredbe Pri tome, prethodni položaj pokazivača memorije ne sme biti na samoj JR naredbi već na bajtu koji određuje relativnu adresu skoka

**29. U**

Naredba U je invarzna naredbi **O** i obavlja vraćanje pokazivača memorije na vrednost pre skoka Pri tome se pamti samo poslednja naredba **O** dok eventualne prethodne bivaju zaboravljene tako da u tom slučaju naredba **U** nema dejstva

**'30. X**

Naredba X pomera pokazivač memorije na vrednost apsolutnog skoka datog **JP** ili **CALL** naredbom Pri tome pokazivač memorije mora stajati ne na samoj **JP** naredbi već na nižem bajtu apsolutne adrese

**31 V**

Naredba **V** je invarzna naredbi **X** i vraća pokazivač memorije na vrednost pre skoka Pamti se samo skok poslednjom naredbom **X**, dok svi prethodni skokovi bivaju zaboravljeni tako da tada naredba **V** nema dejstva

***Jovan Skuljan***

# Šta ima novo

## Trampa

*Ako imate komodor 64 i živite u Engleskoj) a želite „komodor 128 možete da date svoj stari 64 ovlašćenim prodavnicama i da, za uzvrat dobijete popust od 50 funti pri kupovini komodora 128 Zar 64 vredi samo 25 000 dinara?*

## Amstradova „amiga"?

*Sve se glasnije čuju vesti o skorom završetku nove Amstradove mašine koja treba da ima oko pola megabajte 68000 sve one mogućnosti koje i slične klase (ST amige ) i naravno nižu cenu od konkurencije Možda je to amiga' za obične smrtnike pardon Jugoslovene*

## Mek raste i jača

*Svi oni koji su plakali nad nemoćnim i ograničenim mekom biće silno obradovani kad čuju da je Epl spremio mekintoša +' koji bi imao 1Mb RAMa 128K ROMa i dvostrane disk drajvove od 800K Pominje su čak i neko iznenađenje pod tajnim imenom Karla koje bi imalo 4Mb RAMa i hard disk Eto sna za najvernije makovce*

## Slava teško umire

*Ejkorn je sklopio ugovor sa meksičkom kompanijom Dataum po kome Dataum ima pravo da sklapa i prodaje BBC i obadve Amerike Nešto slično je uradio i na istoku Ejkornov veliki tata Oliveti će prodavati BBC u Istočnoj Evropi Istočna Evropa — zvuči baš neutralno Da nismo možda i mi tu uključeni?*

## Računar u kasi

*Ako se pažljivo zagledaju reklame u svim računarskim časopisima (ne kod nas), lako je primetiti da je jako povećan broj reklama za računare u kitu pogotovu za razne varijante PCa Eto načina da se malo smanje cene!*

## Od bejzika do bejzika

*Basic to basic je jako koristan program za vlasnika Enterprise On omogućava da se direktno u Enterprisov računar unose spektrumovi bejzik programi BTB lede preinačuje spektrum bejzik u enterprajz bejzik i naznačava one delove programa koje korisnik mora sam ponovo da napiše Zgodno?*

## Najbolje stvari su besplatne

*U SAD postoji više hiljada baza podataka koje sadrže informacije koje se kreću u rasponu od cena za sirovu naftu u Dubaiu do celokupnog softvera za IBM PC ali ne postoji baza podataka bar za sada o tome šta koja baza podataka sadrži i koliko staje pristup određenoj bazi Ipak postoji način da dođete do toga kako i uz koju cenu šta možete da saznate iz odgovarajuće baze podataka Dovoljno je da pišete na adresu Information USA 12400 Beal Moutain Rd Potomac Md 20854 (telefon je 301-983-8220) i da vas stave na listu osoba koje svaka tri meseca dobijaju besplatan katalog Computer Data Report koji sadrži spisak svih baza podataka i onih besplatnih (ako se izuzme telefon) i onih za koje se plaća pristup kao i šta dotične baze podataka sadrže Ako ne možete da čekate po 3 meseca za 48 dolara ćete dobijati ažurirane kataloge svakog meseca*

***N. A.***

## Novogodišnji pobednici (i gubitnici)

*Još uvek se ne zna precizno ko je najviše dobio od božićnog i novogodišnjeg buma Veliki okršaj između Atarijevog 520 ST i amige koji su najavljivali svi poznavaoci mikrobiznisa izgleda da je odložen za bolje dane Ovog puta su izgleda dobro prošli i Atari i Komodor (Atari malo bolje) dok je žrtva ispao Epl kojem je drastično opala prodaja familije II što mu je bio glavni izvor prihoda već duže vremena Eto tako smo ušli u 1986 koja će sudeći po početku biti jako zanimljiva za računarstvo*

## Jeftina animacija

*Broderbund Software proizvodi specijalni Fantavision generator koji omogućava da se sa eplom ile naprave animirane scene Isti taj generator omogućava i transformisanje oblika između dva lika i postizanje specijalnih efekata Generator proizvodi i do 64 sličice između dva oblika između kojih vrši animaciju Sve to za 50 dolara*

## Dečja posla

*Evo nečeg za one koji su odrasli uz komplete električnih slagalica Kupite vašoj deci The Robotic Computing Kit koji pruža mogućnost da se sastavi mnogo različitih kompleta među njima i ploter) koje posle možete da kontrolišete vašim računarom*

## Posle PCja AT (pa XT)

*Proizvođačima je već dosadilo da prave mašine kompatibilne sa PCjem pa su sada masovno prešli na savršenije verzije to jest PC AT i PC XT Jedna od takvih je Hewlett-Pickardov Vectra PC koji je kompatibilan sa IBM PC AT Sve je isto samo je sve pomalo drugačije*

## Još jedna kartica

*Cumana je proizvela svoju verziju softverske kartice pod imenom Astron Integrated Circuit Card Očekuje se Pocket Wordstar u džepnoj verziji Za sva značajnija mikroračunare već postoje interfejsi*

# Na tajnom kanalu

# Javni ključ

***Da bi se započela partija pokera telefonom neophodan nam je odgovarajući način šifrovanja i dešifrovanja karata Ovaj način ima i lepo englesko ime public-key protocol, i nešto manje romantičan prevod protokol sa javnim ključem Ovi protokoli su, po mišljenju mnogih naučnika danas glavna i najuspešnija oblast istraživanja teorije informacija, teorije brojeva i računarstva i koriste se masovno da bi se obezbedila tajnost informacija***

Ideja protokola sa javnim ključem je jednostavna Potrebno je konstruisati funkciju koju je lako izračunati u jednom i gotovo nemoguće u drugom pravcu — osim ako vas autor jednosmerne funkcije (a ovakve se funkcije nazivaju jednosmernim) nije snabdeo odgovarajućim informacijama Pogledajmo blag ali poučan primer Data je funkcija

y=x ** 4+12 * x ** 3+40 * x ** 2+24 * x −37 i treba izračunati y za x=10 Lako se dobija da je 26203 Ali kad je dato da je y=26203 nije preterano trivijalno izračunati x Naravno ako se ne zna da je

y=((x+3) ** 2−7) ** 2−41

Kad se to zna treba jednostavno na y dodati 37 izvaditi kvadratni koren dodati 7 opet izvaditi koren i oduzeti tri i dobija se traženo x

## Sa okusom pedagogije

Naravno ovo je pedagoški primer i ni za jednog kriptologa ne bi predstavljao ni najmanju teškoću

Pogledajmo još jedan protokol sa javnim ključem prvo u pedagoškoj a zatim i u znatno jačoj verziji (i ovaj ima lepo englesko ime knapsack u prevodu opet prozaično ranac)

Posmatrajmo vektor E=(1,2 4 8 17,35 68 142)

Osam komponenti ovog vektora čine superrastući niz — svaki sledeći član je veći od zbira svih prethodnih Ako treba da pošaljemo binarnu poruku

D=(1 0 0 1 0 1,1 1)

možemo da je šifrujemo pomoću E kao skalarni proizvod dva vektora

S=E * D=1+8+35+68+142=254

Dekodovanje je jednostavno superrast omogućava da se od S oduzimaju redom, od većih ka manjim komponentama, članovi niza i kad je oduzimanje moguće (ako se oduzimanje smatra mogućim dok je razlika veća od nule) i dobije poruka D

| | | | | |
|---|---|---|---|---|
| 254− 142= | 112 | ) 0 | bit poruke | 1 |
| 112− 68= | 44 | ) 0 | bit poruke | 1 |
| 44− 35= | 9 | ) 0 | bit poruke | 1 |
| 9− 17= | −8 | ( 0 | bit poruke | 0 |
| 9− 8= | 1 | ( 0 | bit poruke | 1 |
| 1 [illegible] 4 [illegible] | 3 | ( 0 | bit poruke | 0 |
| 1− 2= | −1 | ( 0 | bit poruke | 0 |
| 1− 1= | 0 | =0 | bit poruke | 1 |

i, istovremeno kraj poruke

## Šifra u rancu

Naravno E se ne koristi za javni protokol Za ovaj protokol možete koristiti hard knapsack ' (težak ranac) Da bi se generisao ovaj ranac ' biraju se dva tajna prirodna broja M i W tako da je

i) M veće od zbira svih komponenti i vektora E

i i) W i M uzajamno prosti (Što znači da je W ** (−1) * W=1 mod M)

Sledeći čuveni članak R C Merkle and M E Helman „Hiding Information and Signatures in Trapdoor Knapsack, „IEEE Trans Information Theory Septembar 1978 od 525 do 530 stranice, Izaberimo M=291 i W=176 (što znači da je W ** (−1)=167) Sada H generišemo iz E pomoću formule

Hj=W * Ej mod M

gde je malo j oznaka za mesto komponente Tako imamo

H=(176 61,122 244,82 49 37 257)

U idealnom slučaju, H izgleda kao niz slučajnih brojeva a superrast struktura originalnog E je skrivena Da bi se dešifrovalo D koristimo H

Sh=H * D=703 Ako je broj komponenti H veliki recimo nekoliko stotina tada se čak i kad se znaju H i Sh D ne može saznati u nekom razumnom vremenu Onaj kome je poruka upućena lako izračunava D pomoću formule

Se=Sh * W ** (−1)) mod M

Možda vam sva ovo izgleda pomalo komplikovano U tom slučaju, probajte da koristite ovaj primer (ili sledeći ideju primera, konstruišite jednostavniji) na svom računaru ili, čak ručno i ideje će se razjasniti

## Na proste činioce

Pri igranju pokera telefonom treba biti oprezan i izbegavati protivnike koji imaju „kraje ' ili neki drugi računar slične arhitektura i performansi Naime arhitektura ovog računara dozvoljava vrlo efikasno rastavljanje na proste činioce brojeva sa velikim brojem cifara, korišćenjem algoritma kvadratnog sita čiji je autor Karl Pomerans (Karl Pomerance) sa Džordžije univerziteta Ovo rastavljanje, s druge strane omogućava razbijanje i najtežih ranaca

Gustava Simons (Gustavus Simmons) Džems Devis (James Davis) i Dijana Holdridž (Diane Holdridge) iz Los Alamos Laboratorije su rastavljali radom brojeve od 55 58 60 63 67 i 69 cifara, za šta im je bilo potrebno vreme od 1 8 do 32 časa Posle daljeg ubrzanja algoritma za 6 45 časova rastavljen je 71 cifrani broj Ako vas interesuju faktori broja 2 ** 251 −1 to su 503 54 217, 176 230 287 214 063 289 511, 81 876 882 198 695 257 501 367 i 12 070 396 178 249 893 039 989 681 Ovaj broj je ACM stavio na plaketu koju je poklonio IEEE povodom stogodišnjice ovog izuzetno cenjenog društva Bilo je to davne 1984 godine Rad na rastavljenju sve većih i većih brojeva se nastavlja

## Ćaskanja sa Enom

Ako niste montekarlovski nastrojen pojedinac moguće je da ća vas više interesovati neki od sledećih problema

1 Ena iz Zagreba i Mirko iz Beograda razgovaraju telefonom i reše da malo obnove svoja sećanja na lepe dane iz Portoroža Za početak se dogovore da bacaju novčić Kako kad jedno od njih baci novčić drugi bude siguran da ga — kad on kaže na koju je stranu pao novčić — onaj koji je bacao ne prevari i kaže da je pao na suprotnu

2 Ena u Zagrabu ima sef za koji Mirko zna šifru a Mirko sef u Beogradu za koji Ena zna šifru Kako da izmene šifra a da svako od njih bude siguran da ga ono drugo neće prevariti (bar kad je u pitanju šifra za sef)

3 Ena ima sef čiju šifru osim nje znaju Mirko i Mario Kako napraviti algoritam koji omogućava i Mirku i Mariju da pristupe sefu ali tako da Ena uvek tačno zna ko je i koliko puta pristupao sefu Mirko i Mario smeju da znaju samo za sebe a ne i za onog drugog

4 Ena i Mirko traba da potpišu ugovor Kako da naprave proceduru potpisivanja tako da Ena ne potpiše dok nije sigurna da je Mirko potpisao i Mirko ne potpiše dok nije siguran da je Ena potpisala

I na kraju jedan ozbiljniji zadatak o naučnicima

5 Trideset naučnika radi na tajnom projektu, ali dokumentaciji mogu da pristupe jedino ako ih je prisutno šesnaestorica ili više Kako da naprave šifru na sefu sa dokumentacijom tako da je ako ih je manje od 16 pristup nemoguć

Ako su vam se zadaci evidali sve je jasno — kriptografija je nauka za vas Mi vam preporučujemo Kanovu knjigu (Kahn D The Codebreakers New York Macmillan) Ako vam to nije dovoljno pišite nam i mi ćemo vas uputiti na dalju literaturu

***Miodrag Potkonjak***

*Umesto dodatka*

# 'amstradov' kalkulator

Pisanje ıole složenijih programa u mašinskom jeziku ne može se ni zamisliti bez poznavanja rutine koje se nalaze ugrađene u ROM-u računara, kao i adresa pozivanja istih sa početnim i krajnjim uslovima koji moraju biti zadovoljeni. Na žalost, dodatak SVE AMSTRADOVE RUTINE objavljen u „Računarima 10" ne opravdava u potpunosti ovaj svoj naslov, jer ne sadrži rutine koje su neophodne pri matematičkim operacijama u raznim više ili manje složenim programima. Kako ni publikacija „CPC-464 FIRMWARE" po kojoj je i rađen ovaj dodatak ne sadrži njihov opis, novi saradnik „Računara" dr Ing Svetislav Zahar je proanalizirao donji ROM i otkrio rutine koje se ne pominju u „amstradovoj" literaturi.

Koristeći program za disassembliranje „mašinca" sopstvene izrade, autor je snimio i odštampao celokupan donji ROM od adresa &0000 do &3FFF, kao i JUMPBLOCK između adresa &B900 i &BFFF. Kako se skok u donji ROM, gde se nalaze tražene rutine, vrši naredbama RESTARTA, tj. RST1 i RST5, bilo je logično da se adresa poziva nalaze u džamp-bloku sa rastojanjem od tri bajta između dve ulazne adresa. Kako su u pomenutoj publikaciji „CPC-464 FIRMWARE" analizirane rutine do adrese &BD3A, pretpostavio sam da se tražene rutine nalaze iznad ovih adresa. Ova pretpostavka se pokazala tačnom, tako da sam pronašao ne samo pozivne adrese za većinu matematičkih rutine, već i potrebne polazne uslove.

Pošto smatram da ovo može zanimati veći broj korisnika mikroračunara AMSTRAD-SCHNEIDER CPC464, to u daljem tekstu dajem prikaz najvažnijih matematičkih rutina sa adresom poziva, početnim i krajnjim uslovima, kao i efektom koji se tom rutinom ostvaruje. Kako se do ovog saznanja došlo analiziranjem ROM-a, logično je da se simbolička, odnosno skraćena imena ne mogu dati, već su nazivi opisnog tipa.

Mada je ovaj tekst namenjen prvenstvo onima koji su već donekle ušli u tajna mašinskog programiranja, neće biti suvišno da se prikaže način memorisanja realnih brojeva u memoriji računara. Za razliku od celih brojeva koji su predstavljeni sa jednim, odnosno dva bajta, bilo u memoriji računara bilo u registrima, za prikazivanje realnih (decimalnih) brojeva koristi se pet bajta u memoriji raspoređenih po svojoj važnosti u obrnutom redosledu.

Svaki realan broj se može prikazati u obliku proizvoda M*2∧K gde je

M — mantisa broja između 0.5 i 1.0, pozitivna ili negativna, a

K — karakteristika, odnosno eksponent broja 2.

Karakteristika broja uvećana za 128 (odnosno &80) smešta se u peti bajt, dok se ostala četiri koriste za predstavljanje mantise. Pošto se ovde radi o brojevima manjim od jedinice, to se oni, za razliku od celih brojeva, dobijaju kad se vrednost sadržana u bajtu sa najvećom težinom (četvrti bajt) podeljena sa 256 sabera sa vrednošću sledećeg (trećeg) bajta podeljenom sa 256∧2 itd. do zadnjeg bajta po važnosti (prvog u memoriji) koji se deli sa 256∧4. Ovo se najbolje može razumeti iz jednog primera.

Broj 438.76 može se predstaviti u vidu proizvoda

0.856953125 * 2 ∧ 9

tako da se u petom bajtu upisuje karakteristika uvećana za 128, odnosno &89. Mantisa se može prikazati u obliku

219/256 + 97/256 ∧ 2 + 71/256 ∧ 3 + 174/256 ∧ 4

tako da bi sadržaj ostala četiri bajta (od četvrtog do prvog) bio &DB &61 &47 i &AE. Međutim, kako je usvojeno da mantisa bude u granicama između 0.5 i 1.0, tako da je sedmi bajt četvrtog bajta uvek jednak jedinici, to se ona može izostaviti, a na njeno mesto staviti predznak broja — 0 za pozitivne, a 1 za negativne brojeve.

Uzimajući u obzir sve ovo, broj 438.76 se u memoriji računara zapisuje u obliku

&AE &47 &61 &5B &89

Negativna vrednost ovog broja, tj. −438.76, razlikovaće se samo vrednošću četvrtog bajta

&AE &47 &61 &DB &89

## PREMEŠTANJE REALNOG BROJA &BD3D

Realan broj, definisan sa pet bajtova, koji se nalazi u memoriji računara, prepisuje se u novih pet bajtova na željenoj adresi.

Početni uslovi
Registar DE sadrži adresu memorije gde se broj nalazi.
Registar HL sadrži adresu na koju se preslikava taj broj.

Krajnji uslovi
Registar A sadrži eksponent (karakteristiku) broja uvećan za &80. Registri BC, DE i HL ostaju sačuvani.

Efekat
U pet bajtova memorije, čija je adresa u registru HL, prenose se vrednosti pet bajtova sa adresa koja se nalazi u registru DE. Sadržaj izvornih pet bajtova ostaje neizmenjen.

## SMEŠTANJE REALNOG BROJA &BD40

Broj koji se nalazi u registru HL smešta se kao realan broj u memoriji na datoj adresi.

Početni uslovi
Registar HL sadrži broj koji se unosi.
Registar DE sadrži adresu na koju se broj želi uneti.

Krajnji uslovi
Registri HL i DE zamenjuju sadržaj, tako da registar HL umesto broja sadrži adresu gde je taj broj smešten.
Registar AF je zaprljan.

Efekat
Broj iz registra HL, koji je predstavljen sa dva bajta, pretvara se u realan broj definisan sa pet bajtova da bi se sa njim mogle vršiti matematičke operacije, i smešta u naznačenu memoriju.

## PREUZIMANJE BROJA IZ MEMORIJE &BD46

Registar HL preuzima realni broj koji se nalazi na datoj adresi u memoriji računara.

Početni uslovi
Registar HL sadrži adresu gde se nalazi željeni broj.

Krajnji uslovi
Registar HL sadrži celobrojni deo realnog broja iz memorije.
Registri BC i DE ostaju sačuvani, dok je registar A zaprljan.

Efekat
Realni broj koji se nalazi u pet bajtova memorije čija se adresa nalazi u registru HL pretvara se u celobrojnu konstantu, koja se zatim smešta u registar HL. Sadržaj memorije gde se taj broj nalazio ostaje neizmenjen.

## SABIRANJE DVA REALNA BROJA &BD58

Vrednost jednog realnog broja iz memorije dodaje se drugom realnom broju.

Početni uslovi
Registar DE sadrži adresu pet bajtova memorije u kojima se nalazi broj koji se želi dodati.
Registar HL sadrži adresu memorije gde se nalazi realan broj kome se dodaje prethodni broj.

Krajnji uslovi
Registar HL ostaje sačuvan — sadrži adresu rezultata.
Registri AF BC i DE su zaprljani

Efekat.
Realni broj koji je predstavljen sa pet bajta iz memorije čija se adresa nalazi u registru DE sabira se sa realnim brojem čija se adresa nalazi u registru HL. Rezultat se smešta umesto broja čija je adresa određena registrom HL tako da sadržaj ovog registra na kraju operacije sadržava adresu rezultata

## ODUZIMANJE REALNIH BROJEVA &BD5B

Sadržaj jednog realnog broja iz memorije se oduzima od drugog realnog broja koji se takođe nalazi u memoriji računara

Početni uslovi
Registar DE sadrži adresu broja koji se oduzima
Registar HL sadrži adresu broja od koga se oduzima prethodni broj

Krajnji uslovi
Registar HL sadrži adresu memorije gde je smešten rezultat.
Registri AF BC i DE su zaprljani

Efekat
Realan broj koji je predstavljen sa pet bajtova smeštenim na adresi koja se nalazi u registru DE oduzima se od realnog broja čija je adresa data registrom HL a rezultat se smešta umesto umanjitelja (sa adresom u HL). Sadržaj memorije gde se nalazio umanjilac ostaje neizmenjen

## ODUZIMANJE REALNIH BROJEVA &BD5E

Kao i u prethodnom slučaju ova rutine vrši oduzimanje dva realna broja, samo što je uloge registra DE i HL izmenjena

Početni uslovi
Registar HL sadrži adresu memorije gde se nalazi broj koji se oduzima
Registar DE sadrži adresu memorije gde se nalazi broj od koga se vrši oduzimanje

Krajnji uslovi
Registar HL sadrži adresu rezultata.
Registri AF BC i DE su zaprljani

Efekat
Realan broj koji se nalazi u pet bajtova memorija čije je adresa sadržana u registru HL oduzima se od realnog broja čija se adresa nalazi u registru DE. Rezultat se smešta u memoriji gde se ranije nalazio umanjilac. Sadržaj memorije gde se nalazio umenjitelj ostaje neizmenjen

## MNOŽENJE REALNIH BROJEVA &BD61

Realan broj iz memorije množi sa se drugim realnim brojem smeštenim takođe u memoriji računara

Početni uslovi
Registar HL sadrži adresu pet bajtova memorije gde je smešten broj koji se množi
Registar DE sadrži adresu memorija gde je smešten broj sa kojim se prvi broj množi

Krajnji uslovi
Registar HL sadrži adresu memorije gde je smešten rezultat.
Registri AF BC i DE su zaprljani

Efekat
Realan broj koji je predstavljen sa pet bajtova memorije čije je adresa predstavljena sadržajem registar HL množi se sa brojem čije je adresa data u registru DE. Rezultat se smešta umesto prvog broja čija adresa ostaje u registru HL. Sadržaj memorije drugog broja ostaje neizmenjen

## DELJENJE REALNIH BROJEVA &BD64

Realan broj smešten u memoriji deli se sa drugim brojem koji se takođe nalazi u memoriji računara

Početni uslovi
Registar HL sadrži adresu broja koji se deli
Registar DE sadrži adresu broja sa kojim se prvi broj deli



Krajnji uslovi
Registar HL sadrži adresu memorije gde je smešten rezultat
Registar AF BC i DE su zaprljani

Efekat
Realan broj koji se nalazi u pet bajtova memorije čija je adresa definisane vrednošću registra HL deli se sa brojem iz memorije čije se adresa nalazi u registru DE. Rezultat se smešta umesto delitelja dok sadržaj memorije gde se nalazio delilac ostaje neizmenjen

## PROMENA ZNAKA &BD6D

Realnom broju koji se nalazi u memoriji menja se predznak

Početni uslovi
Registar HL sadrži adresu memorije gde je smešten broj

Krajnji uslovi
Svi registri osim registra A ostaju sačuvani

Efekat
Realnom broju koji se nalazi u pet bajtova memorije čija se adresa nalazi u registru HL menja se predznak tj. pozitivni broj se pretvara u negativan i obratno. Adresa tog broja i dalje ostaje u registru HL

## ISPITIVANJE ZNAKA &BD70

Rutina vrši ispitivanja broja u memoriji i vraća predznak broja.

Početni uslovi
Registar HL sadrži adresu gde se broj nalazi

Krajnji uslovi
Registar HL sadrži adresu gde se broj nalazi

Krajnji uslovi
Registar A sadrži kod predznaka i to
A=1 ako je broj u memoriji pozitivan
A=−1 ako je broj u memoriji negativan,
A=0 ako je broj u memoriji jednak nuli
Svi ostali registri ostaju sačuvani

Efekat
Ispitivanje realnog broja koji je smešten u pet bajtova memorije čije se adresa nalazi u registru HL i vraćanje koda predznaka tog broja. Sadržaj memorije ostaje neizmenjen

## KONSTANTA PI &BD76

Smeštanje konstante PI na željeno mesto u memoriji

Početni uslovi
Registar HL sadrži adresu pet bajtova memorije gde se želi smestiti konstanta PI.

Krajnji uslovi
Registari BC i HL ostaju sačuvani
Registar A sadrži eksponent konstante uvećan za &80
Registar DE je zaprljen (sadrži adresu ROM-a gde je konstanta PI smeštena)

Efekat
Prenosi se vrednost konstante PI iz ROM-a u pet bajtova memorije čija je adresa data u registru HL da bi se mogla koristiti pri računanju

## KVADRATNI KOREN &BD79

Nalaženje vrednosti kvadratnog korena realnog broja koji se nalazi u memoriji računara

Početni uslovi
Registar HL sadrži adresu pet bajtova memorija gde je smešten broj čiji se kvadratni koren traži

Krajnji uslovi
Registar HL sadrži adresu rezultata
Ostali registri su zaprljani

Efekat
Izračunavanje vrednosti kvadratnog korena realnog broja koji se nalazi u pet bajta memorije čija je adresa smeštena u registru HL i smeštanje rezultata umesto datog broja

## PRIRODNI LOGARITAM &BD7F

Nalaženje vrednosti prirodnog logaritma realnog broja koji se nalazi u memoriji računara

Početni uslovi
Registar HL sadrži adresu gde se nalazi realni broj čiji se logaritam traži

Krajnji uslovi
Registar HL sadrži adresu memorije gde je smešten rezultat.
Svi ostali registri su zaprljani

Efekat.
Izračunavanje vrednosti prirodnog logaritma (logaritma za osnovu e) realnog broja koji je smešten u pet bajtova memorije čije se adresa nalazi u registru HL. Rezultat se smešta na isto mesto

## DEKADNI LOGARITAM &BD82

Nalaženje vrednosti dekadnog logaritma realnog broja koji se nalazi u memoriji računara

Početni uslovi
Registar HL sadrži adresu broja čiji se logaritam traži

Krajnji uslovi
Registar HL sadrži adresu gde je smešten rezultat.
Svi ostali registri su zaprljani

Efekat
Izračunavanje vrednosti logaritma za osnovu 10 realnog broja koji je smešten u pet bajtova memorije čija se adresa nalazi u registru HL. Rezultat se smešta umesto datog broja

## EKSPONENCIJALNA FUNKCIJA &BD85

Nalaženje vrednosti eksponencijalne funkcije realnog broja koji se nalazi u memoriji računara

Početni uslovi
Registar HL sadrži adresu gde se nalazi realan broj

Krajnji uslovi
Registar HL sadrži adresu pet bajtova u memoriji gde se nalazi rezultat. Svi ostali registri su zaprljani

Efekat
Izračunavanje vrednosti eksponencijalne funkcije e× gde je X realan broj smešten u pet bajtova memorije čije je adresa data u registru HL. Rezultat se smešta umesto datog broja

## SINUSNA FUNKCIJA &BD88

Nalaženje vrednosti funkcije SINUS iz realnog broja koji je smešten u memoriju računara.

Početni uslovi
Registar HL sadrži adresu realnog broja

Krajnji uslovi
Registar HL sadrži adresu gde je smešten rezultat
Svi ostali registri su zaprljani

Efekat
Izračunavanja vrednosti sinusne funkcije realnog broja koji je smešten u pet bajtova memorije čija se adresa nalazi u registru HL. Rezultat se smešta umesto datog realnog broja

## COSINUSNA FUNKCIJA &BD8B

Nalaženje vrednosti funkcije COSINUS realnog broja koji se nalazi u memoriji računara

Početni uslovi
Registar HL sadrži adresu memorije gde je smešten dati broj



Krajnji uslovi
Registar HL sadrži adresu memorije gde je smešten rezultat.
Ostali registri su zaprljani

Efekat
Izračunavanje vrednosti cosinusne funkcije realnog broja koji je smešten u pet bajtova memorije čije se adresa nalazi u registru HL i smeštanje rezultata umesto datog broja.

## FUNKCIJA TANGES &BD8E

Nalaženje vrednosti funkcije TANGES realnog broja koji se nalazi u memoriji računara

Početni uslovi
Registar HL sadrži adresu memorije gde je smešten dati broj

Krajnji uslovi
Registar HL sadrži adresu memorije gde je smešten rezultat.
Ostali registri su zaprljani

Efekat
Izračunavanje vrednosti tangesa realnog broja koji je smešten u pet bajtova memorije čije se adresa nalazi u registru HL. Rezultat se smešta umesto datog broja

## INVERZNA FUNKCIJA TANGESU &BD91

Nalaženje vrednosti funcije ARCUS TANGES datog realnog broja koji se nalazi u memoriji računara

Početni uslovi
Registar HL sadrži adresu memorije gde je smešten dati broj

Kranji uslovi
Registar HL sadrži adresu memorije gde je smešten rezultat
Ostali registri su zaprljani

Efekat
Izračunavanje vrednosti funkcije arcus tanges realnog broja koji je smešten u pet bajtova memorije čije se adresa nalazi u registru HL pri čemu se rezultat takođe smešta umesto datog broja

## FUNKCIJA RND &BD9D

Generisanje vrednosti slučajnog broja.

Početni uslovi
Registar HL sadrži adresu pet bajtova memorije gde se smešta rezultat

Krajnji uslovi
Registar HL je sačuvan dok su ostali zaprljani

Efekat
Iz generatora slučajnih brojeva preuzima se vrednost slučajnog broja i smešta u pet bajtova memorije čije je adresa sadržana u registru HL

## PROMENA PREDZNAKA &BDC7

Menjanje predznaka celog broja koji se nalazi u registru HL

Početni uslovi
Registar HL sadrži vrednost broja kome se menja predznak.

Krajnji uslovi
Registar HL sadrži vrednost broja sa izmenjenim predznakom
Registri BC i DE ostaju neizmenjeni dok je registar A zaprljan

Efekat.
Celobrojne veličine koja se nalazi u registru HL zamenjuje se svojom negativnim vrednošću

***Svetislav Zahar***

Biblioteka knjiga

# Primjena osobnih računala u poduzećima

*Izdavač „Informator" Zagreb, grupa autora H Birolla M Habek A Kliment, K Kliment A Pađen i D Ugarković*

Budući da se sve više smanjuju granice između kućnih računala i onih što spadaju u klasu Personal Computera-a dijelu čitateljstva koje zanima ova problematika dobro će doći knjiga grupe autora „Primjena osobnih kompjutora u radnim organizacijama" U redakciji prof dr Antuna Klimanta zagrebački „Informator" potkraj prošle godine tiskao je jednu od rijetkih teoretsko-praktičnih knjiga s tom tematikom u nas, a pošto automatizirano upravljanje poslovnim procesima u uredskom poslovanju jeste preduvjet uspješne proizvodnje te plasmana roba i usluga ovo štivo može pomoći upravljačkim strukturama da se upoznaju s osnovama informatike

Knjiga sadrži osam poglavlja Mjesto i uloga osobnih kompjutora u poslovanju poduzeća Proizvođači osobnih kompjutora Arhitektura osobnih kompjutora. Software-ska podrška, Programiranje na osobnom kompjutoru Tehnologija obrade poslovnog teksta Tehnologija obrade podataka i Poslovni informacijski sustav

U prvom poglavlju autori razmatraju značaj automatizacije u poslovanju poduzeća razvoj i primjenu kompjutora Već tu naglašavaju razliku između već klasične obrade podataka na velikim sistemima i primjenu malih računala u poslovanju poduzeća Ako su potrebne dodatne informacija na primjer korisnici PC-a su u prednosti jer mogu jednostavno komunicirati s računalom bilo da su na radnom mjestu u kući ili na putu

Drugo poglavlje obuhvaća pregled nekih domaćih i inozemnih proizvođača računala Najviše pažnje autori su poklonili Iskrinom modelu „Partner" i PC-u M-11 što ga proizvodi zagrebački Infosistem-Tehničar

Ograničavanje domaće proizvodnje na samo dva modela, vjerojatno iz komercijalnih razloga ili žurba ujedno je i najveća mana u dosta opširnom izlaganju (knjiga ima 236 stranica) Od domaćih proizvođača spomenuti su PEL „Ivo Lola Ribar" Ei Niš TRS Zagreb „Velebit" Novkabel Infosistem-tehničar i Iskra-Delta izostavljeni su „Školski servis" Zagreb Ivasim-elektronika Informatika i elektronika Javor Bitola „Jožef Stefan" Ljubljana Elektronika-inženjering Zemun, Međimurjeoptet Čakovec kao kooperant RIZ — „Nixdorf Micronic" Zagreb i „ZEMA" Austrija UNIS i drugi

Od inozemnih proizvođača prikazani su IBM i Apple

Arhitektura osobnih računala detaljno je opisana a posebni prilog u boji može pripomoći onima koji nisu nikada vidjeli računalo štampač diskete i drugo

Četvrto poglavlje donosi informacije o softverskoj podršci osobnih računala Softver je podijeljen u dvije grupe u aplikativni i sistemski software Sistemski je pak raščlanjen na operacijske sustave prevodioce simboličkih programskih jezika i pomoćne programe Aplikativni softver podijenjan je na onaj koji razvija i programira sam korisnik i aplikativne programe drugih autora koje korisnik kupuje Obrađeni su i programski jezici ADA ALGOL APL BASIC i sedam drugih programskih jezika uz dva pomoćna programa EDITOR i SORT

U petom poglavlju obrađeno je programiranje na osobnom kompjutoru Potanko je razrađen način programiranja u bejziku kao najčešće korištenom programskom jeziku

Za poslovne ljude i sve one koji žele računalo koristiti za obradu teksta naročito će biti korisno šesto poglavlja gdje je iznesena tehnologija obrade poslovnog teksta Autori se ne bave samo načinima formiranja teksta već donose i obrasce kako treba izgledati poslovno pismo kako se može kreirati poslovna komunikacija kako se mogu obraditi pomoću kompjutora masovne poslovne komunikacije s naglaskom na komercijalnim poslovnim informacijama Nastavno u sedmom poglavlju o tehnologiji obrade podataka objašnjava se primjena „personal computinga" na modelu IBM-a

Poslovni informacijski sustav tema je osmog poglavlja gdje se kibernetskim pristupom nude rješenja za komuniciranje u poslovnom informacijskom sustavu Pri tome su se autori pridržavali postojećih zakonskih normi i standarda tako da je model izrade poslovnih informacija primjenljiv u privredi

Kućni ili hobi kompjutori danas su sredstvo za rad za igre i učenje i bit će sredstvo komuniciranja krajem ovog stoljeća Osobni kompjutori preuzeli su od kućnih hobi igre pa se osim za posao PC može koristiti i za zabavu u slobodno vrijeme U nas je doduše, odviše velika razlika u cijeni između PC-a i malog računala ali tako neće vječno i ostati Stoga „Primjena osobnih kompjutora u radnim organizacijama" predstavlja još jedan vrijedan pokušaj da se proširi informatičko znanje i da se i poduzeća uključe u svjetski proces primjene malih i poslovnih računala

Knjiga je tiskana u nakladi od 2 000 primjeraka u već provjerenoj „Informatorovoj" standardnoj opremi

*B. Hebrang*

# operacija FILL

Prvi susret sa naredbom **FILL** vlasnici „spektruma" su imali kada se pojavio BETA BASIC Među brojnim naredbama i funkcijama, koje donosi ovo proširenje bejzika našla se tu i operacija za popunjavanje zatvorenih slika nazvana jednostavno **FILL**. Na ulazu naredba zahteva dva parametra *y* i *y* koordinatu tačke od koje počinja popunjavanje U rezultatu, čitavo polje oko zadate tačke biće popunjeno, sve dok se ne dostigne granice oblasti, ili ivica ekrana

## *Kako to radi beta bejzik*

Reč *popunjavanja* ovde ima sasvim konkretno značenje *setovanje odgovarajućih bitova u video memoriji* na isti način kako to radi naredba **PLOT.**

S druge strane imamo operaciju brisanja popunjenih figura što bi se iz bejzika moglo izvesti uzastopnom primenom naredbe **PLOT OVER 1**, . BETA BASIC u tu svrhu koristi takođe naredbu **FILL**, ali sa atributom **PAPER** (obična **FILL** naredba inače podrazumeva **FILL INK**)

Ako želimo na primer da na svetlo plavoj pozadini ekrana imamo veliki crveni krug sa manjim zelenim krugom u unutrašnjosti, ukucaćemo recimo ovakav program

```
10 PAPER 5 INK 2 CLS
20 CIRCLE 128 88 80
30 FILL 128 88
40 CIRCLE PAPER 4 OVER 1, 160 88 30
50 FILL PAPER 4,160 88
60 PAPER 7 INK 0
70 STOP
```

U liniji 10 postavlja se plava boja pozadine i crvena boja za tekst Zatim se crta i popunjava crvena kružnica (linije 20 i 30) Zelenom bojom podloge crta se manja kružnica (linija 40) a njena unutrašnjost se zatim briše **FILL** naredbom u liniji 50 Pre završetka programa boje se vraćeju na uobičajenu crno-belu kombinaciju

## *. . i kako bismo to mogli sami*

Program koji simulira naredbu **FILL** mogao bi se svakako izvesti i iz bejzika ali bi to bilo neprihvatljivo sporo rešenje Jedini izlez je očigledno sastaviti odgovarajuću mašinsku rutinu i nekako je povezati sa bejzikom da bi korišćenje nove operacije bilo koliko-toliko jednostavno

Pred nama su prema tome dve problema sastaviti algoritam i mašinski program za operaciju **FILL** i povezati taj program sa bejzikom na prihvatljiv način pri čemu u prihvatljive načine ne ubrajamo prenošenje parametara PEEK-POKE metodom

Pre svega moramo videti koliko ulaznih parametara program treba da ima Dva su obavezna (*x* i *y* koordinata polazne tačke na ekranu) a što se tiče izbora boje najjednostavnije je zadati samo parametar koji definiše **FILL PAPER** ili **FILL INK**, dok se sama boja može određivati i unapred standardnim naredbama **PAPER** i **INK**

# Majstorije na računaru/spektrum

***Za Sinklerov bejzik ima uvek argumenata i za i protiv. Međutim, izvesno je da taj jezik ne podržava u adekvatnoj meri grafičke mogućnosti računara, koje, uprkos ograničenju u radu sa bojama (najviše dve boje u istom znakovnom polju), nipošto nisu beznačajne: rezolucija 256×192 predstavlja i te kako velik potencijal — samo ga treba na pravi način iskoristiti. U tom cilju, svakako, moramo raspolagati širim izborom naredbi. Instrukcije: PLOT, DRAW i CIRCLE, koje „spektrum" jedino raspoznaje, mogu služiti tu samo kao osnova za izgradnju potpunijeg sistema. U ovom tekstu ćemo posvetiti pažnju naredbi FILL.***

Odlučili smo se da ulazne parametre prosleđujemo metodom **DEF FN** funkcije o čemu smo u „Računarima" već više puta pisali. Recimo ako mašinski program **FILL** počinje na adresi 50000 ukucaćemo u bejziku liniju

```
10 DEF FN f(x y z)=USR 50000
```

Prva dva parametra funkcije **FN** f predstavljaju *x* i *y* koordinatu početne tačke a treći parametar može uzeti vrednost 0 u slučaju **FILL PAPER,** ili 1 u slučaju **FILL INK.** Tako na primer ako želimo da popunimo **(FILL INK)** neku figuru sa početnom tačkom (100 50) izvršićemo

**RANDOMIZE FN f(100,50,1)**

Istu figuru izbrisaćemo sa

**RANDOMIZE FN f(100,50,0)**

Što se tiče onog primera u BETA BASIC-u sa crvenim i zelenim krugom sada bismo ga mogli rešiti i ovako

```
10 DEF FN f(x y z)=USR 50000
20 PAPER 5 INK 2 CLS
30 CIRCLE 128 00 00
40 RANDOMIZE FN f(128 00,1)
50 PAPER 4
60 CIRCLE PAPER 4 OVER 1,100 88 30
70 RANDOMIZE FN f(160 88 0)
80 PAPER 7 INK 0
90 STOP
```

Očigledno, nikakvih suštinskih izmena nema osim što umesto **FILL PAPER 4,** . moramo neposredno izvršiti **PAPER 4** ispred naše **FILL** instrukcije

Ono što sada preostaje da se uradi jeste sastaviti program **FILL** koji će raditi ovako kako smo opisali

### *Jedan loš ...*

Nismo se upuštali u analizu algoritma koji koristi BETA BASIC ali se već na prvi pogled čini da on radi prilično čudno kad jednom popuni celu figuru program provodi iznenađujuće dugo vreme proveravajući da li je obavio sve kako treba ili je možda negde preskočio neku „rupu" (?!) Ako među našim čitaocima ima takvih koji su ovaj program analizirali pozivamo ih da nam se jave i objasne u čemu je stvar

Mi ćemo međutim, predložiti jedan drugačiji algoritam koji je dovoljno jasan i prihvatljiv da ga ovde možemo detaljno opisati Poći ćemo pre svega od formulacije samog problema

Data je neka oblast ograničena proizvoljnom krivom linijom ili ivicom ekrana Data su koordinate *(x y)* proizvoljne tačke u toj oblasti Zadatak je popuniti sve tačke obuhvaćena datom figurom polazeći, pri tome od zadate tačke *(x,y)*

Ograničićemo se sada samo na **FILL INK,** mada je sve potpuno isto i u slučaju **FILL PAPER,** s tom razlikom što je oblast, zapravo već popunjena, a granicu čine prazna polja Tu je zadatak *obrisati* sve tačke u oblasti zadate sve dok se ne dosegne granica

Osnovna teškoća u svemu ovome je *kako pronaći sve tačke* unutar zatvorena oblasti

Jedan vrlo jednostavan postupak bio bi sledeći

Kada god se otkrije nova tačka ispita se i osam tačaka koje je okružuju Svaka prazna od tih osam predstavlja novu tačku oblasti, i njene koordinate se mogu upamtiti Postupak se ponavlja tako što se stalno uzima poslednje upamćene tačke i vrši testiranje osam okolnih Sve će tačke biti pronađene ali po cenu ogromnog utroška memorije praktično ća se u isto vreme čuvati koordinate svih tačaka oblasti!

Od ovakvog algoritma jasno treba odustati Mi smo ga pomenuli jedino zato što se iz njega vidi koliko postavka problema utiče na postupak rešavanja Jer sve vreme govoreći o naredbi **FILL**, mi smo imali na umu popunjavanje *tačku po tačku* A ako malo razmislimo videćemo da je sasvim izvodljivo i popunjavanje *liniju po liniju* horizontalno ili vertikalno

### *... i jedan dobar algoritam*

Tako dolazimo na pravi algoritam

Od zadate tačke povući ćemo levo i desno horizontalnu liniju do granice oblasti Upamtićemo koordinate početne tačke linije i ukupan broj tačaka u liniji (dužinu)

U ciklusu zatim uzimamo poslednju upamćenu liniju krećemo se duž nje tačku po tačku i gledamo stanje ispod i iznad nje Kad god otkrijemo prazninu povlačimo horizontalnu liniju i pamtimo njen početak i dužinu

Postupak se ponavlja sve dok stek na kome čuvamo linije ne ostane prazan Utrošak memorije je minimalan, jer se na steku istovremeno nalaze jedna do dve linije ukoliko se radi o jednostavnoj oblasti, bez mnogo krivina i „zaliva Kod složenijih oblasti broj linije koje se čuvaju opet ne može biti veći od broja „zaliva

Ostaje nam, sada da ovaj opis algoritma prevedemo u mašinske naredbe

Objavljujemo kompletan program sa komentarima namenjen onima koji iz svega ovoga žele nešto da nauče Čitaoci koje više interesuje sama primena programa treba jednostavno da učitaju asembler (npr DEVPAC GENS) i onda pažljivo ukucaju svaku liniju iz priloženog listinga, preskačući naravno komentare Instrukcije **ORG** se može po volji menjati ali se tada mora korigovati i adresa iza funkcije **USR** u odgovarajućoj **DEF FN** naredbi u okviru bajzika

Posle asembliranja program zauzima 179 bajtova Treba ga snimiti na kasetu naredbom **SAVE „FILL" CODE 50000 179** i kasnije ga učitavati sa **LOAD "CODE**

Program **FILL,** u poređenju sa odgovarajućom naredbom BETA BASIC-a radi nešto sporije kada su u pitanju manje figure Recimo u slučaju kruga radijusa 70, izvršenje traje oko 6s prema 5s koliko to radi BETA BASIC Međutim pri popunjavanju većih polja BETA BASIC daleko zaostaje Umesto 34 s koliko je potrebno BETA BASIC-u za popunjavanje čitavog ekrana, programu FILL je dovoljno samo 18 s

Ostale mogućnosti oba programa su međusobno jednaka Ograničenje jedino stvara loša „spektrumove" rezolucija u boji, tako da bojenje bilekih figura može da nam pričinjava probleme

***Jovan Skuljan***

## *Zbrka u matrici*

U tekstu „Matrice i nizovi" objavljenom u „Računarima 11" došlo je do niza omaški tehničke prirode zbog kojih dugujemo izvinjenje našim čitaocima

Kada je broj već bio izašao iz štampe ispostavilo se da smo jedan od priloženih programa stavili na svom prirodnom mestu u tekstu Svi ostali su razbacani bez ikakvog reda pa je čitalac u situaciji da traži unaokolo šta mu zatreba Srećna okolnost je što svaki primer ima ime u prvoj **REM** liniji, tako da je traženje olakšano Manjih probleme će jedino biti sa potprogramom za sistem jednačina **N×N** koji je jednim spretnim potezom makazama rasečen na dve dela Prvo ćete naići na deo od linije 3050 a tek pri kraju teksta na ono što tome prethodi I konačno, potpuno je ispušten program 2000 za računanje determinante **N×N** Redakcija se izvinjava čitaocima i autoru teksta

# Program: FILL

### *Autor Jovan Skuljan*

J Definicije simboličkih adresa. Prvih šest se odnosi na rutine iz ROM-a, a preostale tri su standardne sistemske promenljive. Prvi bajt sistemske promenljive MEMBOT (&5C92) program **FILL** koristi kao privremenu memorijsku lokaciju, što ne ometa rad računara, jer kalkulator ne koristi memoriju van bejzik naredbe u kojoj je pozvan.

```
FINDII   EQU   &1E94    Rutina FIND_INT1
TEST_R   EQU   &1F05    Rutina TEST_ROOM
P_ADDR   EQU   &22B1    Ulaz u rutinu PIXEL_ADD
GET_BC   EQU   &2307    Rutina STK_TO_BC
REP_B    EQU   &24F9    REPORT_B Integer out of range.
STKNUM   EQU   &33B4    Rutina STACK_NUM
DEFADD   EQU   &5C0B    Sistemska promenljiva.
ATTR_P   EQU   &5C8D    Sistemska promenljiva.
MEMBOT   EQU   &5C92    Sistemska promenljiva.
```

2 Početak. Prikupljanja ulaznih parametara. Koordinate *x* i *y* će dinakino biti unete u registre C i B, a indikator **PAPER/INK** u varijablu MEMBOT pripremljen za kasniju upotrebu od strane potprograma **PLOT**. Na mašinski stek se postavlja oznaka za dno steka, povlači se početna linija i ulazi se u glavnu petlju.

```
         ORG   50000
FILL     LD    HL,(DEFADD)   U HL se uzima adresa prvog parametra
         INC   HL            funkcije FN I i pomera se na prvi bajt
         INC   HL            binarnog zapisa.
         CALL  STKNUM        X-koordinata ide na računski stek.
         INC   HL            Preskoči zarez
         INC   HL            i pomeri se na početak drugog
         INC   HL            parametra funkcije FN I
         CALL  STKNUM        Y-koordinata ide na računski stek.
         INC   HL            Preskoči zarez,
         INC   HL            i pomeri se na početak trećeg
         INC   HL            parametra funkcije FN I
         CALL  STKNUM        Indikator PAPER/INK ide na stek
         CALL  FINDII        a odatle odmah u akumulator
         CP    &02           Da li je veći od jedan?
         JP    NC,REP_B      Ako jeste, to je greška.
         DEC   A             Posle dve jednostavne operacije indi-
         XOR   &01           kator postaje &01 za INK a &FE za PAPER
         LD    (MEMBOT),A    Sačuvaj indikator za kasnije
         CALL  GET_BC        X-koordinata u C a Y u B
         LD    A,175         Maksimalna vrednost za Y
         SUB   B             Da li je Y van opsega?
         JP    C,REP_B       Ako jeste, prijavi grešku
         EXX                 Pređi u alternativni set registara.
         PUSH  HL            Sačuvaj HL za povratak u bejzik.
         LD    BC,&0000      Pripremi BC za kasnije
         EXX                 Vrati se u osnovni set registara
         LD    HL,RETADD     Adresa RETADD biće iskorišćena za
         PUSH  HL            povratak u bejzik.
         LD    B,A           Invertovano Y (tj. 175-Y) ide nazad u B.
         LD    H,&FF         Signal za dno mašinskog steka, nijedna
         DEC   SP            tačka ekrana nema Y=255. SP se umanjuje
         PUSH  HL            jer stek treba da sadrži tri bajta.
         CALL  DRAW          Iscrtaj početnu liniju, ako je moguće
                             i postavi na stek njene parametre.
```

3 Glavna petlja za ispitivanja horizontalnih linija upamćenih na mašinskom steku. Tri poslednja bajta na steku određuju koordinate početne tačke i dužinu linije. Ali, ako je Y=255, to je signal da je operacija **FILL** privedena kraju.

```
EACH_L   POP   BC            Koordinate početne tačke idu u BC
         DEC   SP            Stek sadrži tri bajta.
         POP   DE            Dužina linije ide u D
```

4 Petlja za ispitivanja svake tačke linija. Gleda se ispod i iznad i povlači nova linija, ako je potrebno.

```
EACH_P   INC   B             Pomeri se na liniju ispod tekuće
         RET   Z             ali, ako je to bilo dno steka, vrati se
         LD    A,B           Testiraj novu Y koordinatu
         SUB   176           Da li je ispod dna ekrana?
         CALL  NZ,DRAW       Pozovi DRAW samo za Y u okviru ekrana.
         DEC   B             Vrati se na tekuću liniju
         JR    Z,NEXT_C      Ako je Y=0, onda nema linije iznad.
         DEC   B             Pređi na liniju iznad tekuće
         CALL  DRAW          Povuci liniju, ako treba.
         INC   B             Vrati se na tekuću liniju
NEXT_C   INC   C             Pređi na sledeću tačku tekuće linije.
         DEC   D             Umanji brojač dužine
         JR    NZ,EACH_P     Ako ima još tačaka, vrati se u petlju
         JR    EACH_L        Vrati se, uzmi sledeću liniju
```

5 Povratak u bejzik. Ispred **RET** naredbe obnavlja se sadržaj HL.

```
RETADD   POP   HL            Obnovi HL.
         EXX                 Zameni registre.
         RET                 Vrati se u bejzik
```

6 Potprogram DRAW proverava tačku sa koordinatama BC i crta liniju levo i desno samo ako je tačka prazna (slučaj **FILL INK**) ili samo ako je popunjena (slučaj **FILL PAPER**). Parametri nove linije (početna koordinata i dužina) stavljaju se na mašinski stek u vidu tri bajta: u vrhu je X, odmah ispod Y, a u dnu dužina.

```
DRAW     LD    H,&00         Brojač dužine linije stavi na nulu
         CALL  PLOT          Nacrtaj tačku, ako je moguće.
         RET   NC            Vrati se ako je tačka već postojala.
         LD    E,C           Sačuvaj X privremeno
FIND_S   LD    A,C           Traži levu granicu
         AND   A             Da li je to leva ivica ekrana?
         JR    Z,EXIT_S      Ako jeste, idi napred na izlaz.
         DEC   C             Umanji X za jedinicu
         CALL  PLOT          Crtaj tačku, ako je moguće
         JR    C,FIND_S      Ako granica nije nađena, traži ponovo
         INC   C             X-koordinata početne tačke.
EXIT_S   LD    L,C           Prenesi X u registar L.
         LD    C,E           Obnovi polaznu vrednost X.
FIND_E   INC   C             Traži desnu granicu
         JR    Z,EXIT_E      Idi napred ako je iza ivice ekrana
         CALL  PLOT          Crtaj tačku, ako je moguće
         JR    C,FIND_E      Ako granica nije nađena, traži ponovo
EXIT_E   LD    C,L           Vrati X-koordinatu u C.
         EXX                 Sačuvaj sve registre
         CALL  TEST_R        Testiraj memoriju (BC =40000).
         EXX                 Obnovi registre.
         EX    (SP),HL       Razmeni dužinu linije i povratnu adresu
         INC   SP            Dužina linije je samo jedan bajt
         PUSH  BC            Početna tačka ide na stek
         LD    C,E           Obnovi polaznu X-koordinatu
         JP    (HL)          Vrati se iz potprograma
```

7 Potprogram PLOT popunjava ili prazni tačku sa koordinatama BC i uvećava pri tome brojač tačaka H. Takođe modifikuje odgovarajući atribut za boju i vraća se sa setovanim indikatorom prenosa (C =1). Međutim, ako je tačka već bila na svom mestu, ništa se ne preduzima i povratak se vrši sa C =0.

```
PLOT     PUSH  HL            Sačuvaj HL
         PUSH  DE            Sačuvaj DE
         LD    A,B           Uzmi Y-koordinatu
         CALL  P_ADDR        Nađi adresu bajta u video memoriji
         LD    E,B           Sačuvaj Y privremeno
         LD    B,A           Položaj tačke u bajtu
         INC   B             Priprema brojača.
         LD    A,(MEMBOT)    Indikator PAPER/INK (&FE/&01)
ROT      RRCA                Priprema maske uzastopnom rotacijom
         DJNZ  ROT           udesno potreban broj puta.
         LD    B,E           Obnovi Y-koordinatu
         LD    E,A           Sačuvaj masku privremeno
         OR    (HL)          Setuj zadati bit u video bajtu
         BIT   7,(IY+88)     Testiraj indikator u MEMBOT
         JR    Z,TEST        Idi napred ako je u pitanju FILL INK
         LD    A,E           Obnovi masku
         AND   (HL)          Resetuj zadati bit u video bajtu
TEST     CP    (HL)          Da li je tačka već bila prisutna?
         JR    Z,FINISH      Ako jeste, idi napred.
         PUSH  AF            Sačuvaj indikatore
         LD    (HL),A        Postavi video bajt na svoje mesto.
         LD    A,H           Nizom operacija
         RRCA                izračunaj adresu
         RRCA                odgovarajućeg
         RRCA                atributa za boju
         AND   &03
         OR    &58
         LD    H,A
         LD    A,(ATTR_P)    Stalni atribut za boju
         LD    (HL),A        Kopiraj zadati atribut
         POP   AF            Obnovi indikatore
FINISH   POP   DE            Obnovi DE
         POP   HL            Obnovi HL
         RET   Z             Vrati se ako je tačka već postojala.
         INC   H             Uvećaj brojač tačaka
         SCF                 Setuj indikator prenosa
         RET                 Vrati se iz potprograma.
```

*Majstorije na računaru amstrad*

# kraljevstvo za registar

***Svakom mašinskom programeru se (ne jedanput) oteo nešto slično naslovu ovog članka. Nedostatak registara je večita patnja svih koji su osuđeni na osmobitne mikroprocesore, a želja da se što više približe mašini na kojoj rade. I pored prave navale 16-bitnih procesora, koji su sve jeftiniji i kod kojih ima mnogo registara, njihovi osmobitni preci još nisu rekli poslednju reč. Još uvek se sve više stvari može na njima uraditi na sasvim zadovoljavajući način i još će mnogo vode, nafte i svega ostalog proteći našom Savom, a oni neće sići sa scene. Tako nešto potvrđuje i pojava računara „amstrad". Na njemu je sve staro bar po pet godina, a nema savremenijeg osmobitnog personalca od njega. Ni procesor nije izuzetak — koristio ga je još TRS 80. Nevolje je jedino u tome što je upotreba alternativnog seta registara na „amstradu" potpuno zabranjena. Nema, međutim, toga čemu pravi programer ne može da doskoči, pa čak i ovakvim ograničenjima.***

Može se slobodno reći da je kod nas najpopularniji mikroprocesor Z80 Tu popularnost su mu, pre svih doneli Sinklerovi računari ZX81 i ZX82 (pardon ZX spektrum ) Većina domaćih hakera je odnegovana upravo na starom dobrom spektrumu" A hakeri najviše vole dobar asembler i po mogućstvu da je gluvo doba Na nesreću, programeri koji rade na mašinskom jeziku osmobitnih mašina su upravo zbog nedostatka registara pravi siromasi i njihovo vreme je najjeftinije ako znate šta se pod tim podrazumeva Prava drama međutim nastaje kada tvorci operativnog sistema zabrane upotrebu određenih registara

Amstradovi računari koriste Z80 On u odnosu na 6502 koji radi sa samo tri registra ima svu silu registara u svakom trenutku dostupno je 7 osmobitnih registara opšte namene A BC DE i HL) dva šesnaestobitna indeksa registra (IX i IY) i redovni statusni registar (F) brojač naredbi (PC) i pokazivač steka (SP) Pored ovoga postoje i alternativni registarski set i dva flip-flopa koji određuju koji set registara se u datom trenutku koristi Logično ti flip-flopovi se kontrolišu pomoću dve naredbe

1) EXX — za registre BC, DE i HL
2) EX AF, AF — za akumulator i statusni registar

Oba registarska seta mikroprocesor tretira na potpuno isti način Među njima jedino čovek (programer) pravi razliku i trebalo bi da u svakom trenutku zna koji od setova koristi pa tako jedan od njih dobija epitet glavnog a drugi alternativnog seta Ako ste dobar programer pa još i dobar strateg ovo je prava stvar za vas Na žalost moramo računati i na



## *Amstradove specifičnosti*

Kod ovog računara su dva pomenuta naredbe asemblera ,zabranjene Naravno radi se o softverskoj mućki programera „lokomotiv softvera izgleda da su imali prilično dobro mišljenje o sebi (ko ga nema) pa su iskoristili alternativni set ljubomorno ga čuvajući samo za svoj operativni sistem Radi se o dosta čudnoj odluci ljudi koji su pravili računar ali je ona ipak motivisana Činjenica je da je mnogo više ljudi čulo za benčmark testova nego za alternativni registarski set Računar se bolje prodaje kada ima bolje benčmark testove i oni se redovno prvi podmeću potencijalnim kupcima (budite sasvim sigurni da se svi prikazi računara tipa , dva-sata-nasamo-sa " započinju gubljenjem pola sata na merenje raznih testova a takvi prikazi često odlučuju umesto vas, iako su neobjektivni i najčešće netačni) Pošto je na žalost novac vrhovni arbitar na zapadu doneta je odluka u korist brzine dobijene na pomalo veštački način, a na štetu programera

Alternativni registarski set se na amstradu ne sme koristiti iz dva razloga prvo — krahiraće naš program koji ga koristi jer OS ne ,čuva registre alternativnog seta i drugo — krahiraće OS jer u nekim od tih registara „drži" svoje promenljive Nemojte pomisliti da to može da se reši izbegavanjem rutina operativnog sistema — obrada interapta do koga dolazi 300 puta u sekundi pripada kernelu a upravo on brije po alternativnom setu Zbog toga se u interapt rutini , amstrada' instrukcija PUSH i POP ne mogu naći ni pomoću lupe Da nije primenjen ovaj način da se prilikom prekida sačuvaju sistemi amstrad bi bio sporiji bar za 10 procenata a time bi bila izgubije-

na svaka šansa da se sustigne BBC B Ono što je Ejkom pružio vlasnicima svojih računara kvalitetnim hardverom to je Amstred izveo softverski na prilično nepopularan način Tako nam sada jedino preostaje da sami softverskim zakrpama pokušamo da rešimo ono što je trebalo rešiti mnogo mnogo ranije

## Pustio bih ja njega, ali

Verovatno je već mnogima pala na pamet ideja da se prekidi onemoguće dok se koristi alternativni set To je ipak više nego polovično rešenje Nije preporučljivo provoditi veći deo vremena bez interapta — tastatura će biti „gluva" neće biti merenja vremena, itd Pojedini programeri (uglavnom oni iskusniji) ne preporučuju iz istih razloga ni upotrebu instrukcije tipa LDIR, ako se radi o većim količinama podataka

Na početku je najbolje da upoznamo način na koji OS koristi alternativni set Za smeštanje promenljivih koriste se samo BC' (alternativni BC registar) i keri fleg U B' registru čuva se adresa I/O porta video vrata (Video Gate Array) — #7F To je vrlo zgodno mesto u kome se odvijaju mnoge značajne stvari ali ostavimo to za neki prikladniji trenutak U C se nalazi vrednost koja se šalje na pomenuti port pa pogledajmo kakva značenja može imati

bit 7,6 — određuju tip komande, u ovom slučaju to će biti, respektivno 10
bit 5 — 0
bit 4 — ovaj bit je u vezi sa interaptom i biće objašnjen u drugom napisu (sisti 0)
bit 3 — kontrola gornjih ROM-ova (1 znači isključi)
bit 2 — kontrola donjeg ROM-a (1 znači isključi)
bit 1 0 — određuju mod u kom se računar nalazi na sledeći način
00 — mod 0 (16 boja 160×200 tačaka)
01 — mod 1 (4 boje 320×200 tačaka)
10 — mod 2 (2 boje 640×200 tačaka)
11 — ilegalna vrednost (ne koristiti)

Alternativni keri fleg (carry') je obično 0 a kada je setovan znači da je nastupio interapt

Preostale registre alternativnog seta kernal koristi da bi izbegao operacije sa stekom (PUSH i POP) koje su prilično spore Na taj način značajno su ubrzani pozivi rutina operativnog sistema preko glavnog džamp-bloka

Pokušajmo sada uz jedan primer da objasnimo upotrebu alternativnog seta registara Primer je gotovo trivijalan ali poslužiće svrsi

Ukoliko ste čitali prethodna „Računare", sigurno se sećate zbrke sa ROM-ovim i RAM-om Kada je neki od ROM-ova „uključen", „čitanje" sa lokacija na kojoj se preklapaju RAM i ROM vrši se iz ROM-a Ponekad programer nije siguran kakvo je stanje nekog ROM-a, a samim tim i odakle će dobiti podatak Za takve slučajeve može se napraviti rutina koja uvek „čita" podatke iz RAM-a. Recimo da ta rutina simulira instrukciju LD A (BC)

```
iz_BC_u_A
DI            ,interapt bi napravio zbrku
EXX           iskoristićemo alternativni
              set
LD A, C       ,u C se nalazi vrednost koju
              možemo poslati na port
              #7FXX
SET 2 A       , isključi donji
SET 3, A        i gornji ROM
OUT (C) A     seda ih stvarno isključi
EXX           adresa je u BC
LD A (BC)     ,pročitaj
EXX
```

```
Program 1:
    usr_irt:  CALL rym_alt
              RST  #38
              JP   rsr_alt
;
    syr_rit:  DI
              EX   AF,AF'
              EXX
              LD   (rrr_HL),HL
              LD   (rrr_BC),BC
              LD   (rrr_BC),BC
              PUSH AF
              POP  HL
              LD   (usr_AF),HL      ;sačuvaj alternativni
                                     set korisnika
              LD   HL,(rym_irt)
              LD   (#39),HL          ;[illegible] adresu,
              LD   BC,(ryr_BC)       ;[illegible] BC
              OR   A                 ;i keri [illegible]
              EXX
              EX   AF,AF'
              RET
;
    usr_alt:  DI
              EX AF AF
              EXX
              LD   (ryr_BC),BC       ;sačuvaj [illegible] BC
              LD   HL,(#39)          ;i adresu rutine [illegible]
              LD   (rym_irt),HL       obrade interapta
              LD   HL,urr_irt        ;umesto adresu [illegible]
              LD   (#39) HL           rutine
              LD   HL,(rrr_AF)
              PUSH HL
              POP  AF
              LD   HL,(rrr_HL)
              LD   DE,(usr_DE)       ;restauriši vrednosti
              LD   BC,(usr_BC)        alternativnog seta
              EXX
              EX   AF,AF
              EI
              RET
;
    rym_irt   DEFS 2    ;adresa sistemske rutine
    rym_BC    DEFS 2    ;sistemske promenljive iz BC'
    rrr_AF    DEFS 2    ;vrednosti
    rrr_BC    DEFS 2     korisničkog
    rrr_DE    DEFS 2     alternativnog
    rrr_HL    DEFS 2     seta
```

```
OUT (C), C    vrati staro stanje ROM-ova
EXX           ,prekiopi registre
EI            sada može da nastupi
              prekid
RET
```

Dakle kad god izvršimo instrukciju CALL iz_BC_u_A u A registru dobijamo vrednost RAM ćelije na adresi BC pri čemu će svi ostali registri i flegovi biti sačuvani (zbog poslednjih su korišćene dve instrukcije SET čiji bi ekvivalent bio instrukcija OR %00001100 koja je kraća ali bi poremetila flegove) Ovako otprilike deluju i izgledaju rutine OS-a koje koriste alternativni registarski set (usput identična rutina koja simulira LD A (HL) već se nalazi u računaru, poziva se sa RST #20 i odaziva na ime RAM LAM) Odmah se primećuje zavidna efikasnost pre svega zbog promenljivih koje se nalaze u BC registru

Ova tehnika upotrebe alternativnog seta dobra je samo za ovako jednostavne i kratkotrajne operacije Ukoliko ono što radimo namerava da uzme veći deo procesorskog vremena, moramo se poslužiti nešto komplikovanijom tehnikom koja bi nam pružila veću slobodu — drugim rečima omogućen interapt

## Sa interaptom

Ukoliko naš program duži period vremena (ili sve vreme) koristi alternativni registarski set moraćemo da se pobrinemo da interapt funkcioniše na valjan način To je moguće učiniti jedino umetanjem naše interapt rutine između zahteva za prekidom i same obrade prekida Pošto svaki zahtev indukuje poziv potprograma koji počinje od adrese #38 na tom mestu moramo da ubacimo rutinu koja će sprečiti nastanak gore-opisane zbrke Po uključenju računara na adresi #38 nalazi se jedna džamp instrukcija koja upućuje na neku adresu u RAM-u — tu je potprogram za obradu interapta Ta adresa nije u svim „amstradima" ista jer je OS pretrpeo dosta izmena od modela 464 do 6128 Umesto tog džampa, ubacićemo adresu sopstvene rutine Njen zadatak bi bio da posle izvesnih operacija pozove rutinu za obradu prekida

Izvesne operacije bi bile sledeće

1) sačuvati korisnički alternativni registarski set
2) restaurisati sistemske promenljive u BC registru i keri flegu
3) vratiti adresu koju pri EMS (early-morning start) OS postavlja iza koda za džamp (#C3) na adresu #39
4) pozvati rutinu za obradu interapta

RST #38),

5) opet postaviti naš džamp na adresu #38

6) sačuvati sistemski BC,

7) restaurisati korenički alternativni set

Verovatno će se velikom broju čitalaca korak 3), a samim tim i korak 5) učiniti na prvi i sve ostale pogleda potpuno nepotrebnim Naprotiv oni su od „vitalnog značaja" što bi rekli naši rukovodioci — neka im se grlo nikad ne osuši od promaje Ako se sećate „događaja" koji su opisivani u prošlom broju „Računara" setite se i činjenice da se neki od njih normalni asinhroni)

```
        org   [illegible]
;(zbog lakšeg modifikovanja, kao primer, ova vis se odnosi)
;na restart #18 (FAR CALL) [illegible]
usralt: di
        ex    af,af'
        exx
        ld    (sysbc),bc
        ld    hl,usrlat
        ld    ([illegible]),hl
        ld    hl,usrrok
        ld    ([illegible]),hl
        ld    hl,usrfar        ;[illegible]
        ld    ([illegible]),hl ;[illegible]
        ld    hl,usrcalc
        ld    ([illegible]),hl
        ld    hl,usrlam
        ld    ([illegible]),hl
        ld    hl,(usrall)
        push  hl
        pop   af
        ld    hl,(usrhl)
        ld    de,(usrde)
        ld    bc,(usrbc)
        exx
        ex    af,af'
        ei
        ret
;______________________________________
sysalt: di
        ex    af,af'
        exx
        ld    (usrhl),hl
        ld    (usrde),de
        ld    (usrbc),bc
        push  af
        pop   hl
        ld    (usrall),hl
        ld    hl,(sysla)
        ld    ([illegible]),hl
        ld    hl,(sysrst)
        ld    ([illegible]),hl
        ld    hl,(sysfar)      ;[illegible]
        ld    ([illegible]),hl ;[illegible]
        ld    hl,(calcret)
        ld    ([illegible]),hl
        ld    hl,(syslam)
        ld    ([illegible]),hl
        ld    bc,(sysbc)
        xor   a
        exx
        ex    af,af'
        ret
;______________________________________
usrhl:  defs 2
usrde:  defs 2
usrbc:  defs 2
usrall: defs 2
sysbc:  defs 2
sysrst: defs 2
sysrs:  defs 2
sysla:  defs 2
calcret: defs 2
;______________________________________
usrlat: call sysall
        rst   [illegible]
        jp    usrall
;______________________________________
usrrok: call sysall    ;prebaci na sistemsku varijantu
        exx            ;sada možemo da koristimo hl bez bojazni
        ld    hl,usrall
        ex    (sp),hl  ;na steku se nalazi adresa kao inline parametar
        push  hl       ;dakle, prvo usrall, pa adresa-parametar na steku
        exx            ;ponovo na glavni set da ne bi došlo do zbrke
        jp    #08      ;skok na restart (povratak će biti na usrall)
;______________________________________
usrfar: call sysall    ;[illegible]
        exx            ;[illegible]
        ld    hl,usrall ;[illegible]
        ex    (sp),hl  ;[illegible]
        push  hl       ;[illegible]
        exx            ;[illegible]
        jp    [illegible] ;[illegible]
;______________________________________
usrcalc: call sysall
        exx
        ld    hl,usrall
        ex    (sp),hl
        push  hl
        exx
        jp    [illegible]
;______________________________________
usrlam: call sysall
        rst   [illegible]
        jp    usrall
;______________________________________
end:    call sysalt
        ei
        ret
;______________________________________
start:  ld    hl,([illegible])
        ld    (sysla),hl
        ld    hl,([illegible])
        ld    (sysrst),hl
        ld    hl,([illegible])  ;[illegible]
        ld    (sysfar),hl       ;[illegible]
        ld    hl,([illegible])
        ld    (calcret),hl
        ld    hl,([illegible])
        ld    (syslam),hl
        di             ;ovaj deo je nepotreban [illegible]
        exx
        push  bc
        pop   bc
        res   [illegible]
        exx
        ei             ;ako se poziva vraća iz [illegible]
        jp    usralt
```

obrađuju neposredno pre povratka iz rutine za obradu prekida i niko vam ne garantuje da će trajati manje od tri milisekunde kada se buda javio novi interapt Sada je jasno šta bi se desilo da u tom trenutku na adresu #39 bude adresa rutine koja sledi pravila datog algoritma

Po ugledu na struktuirano programiranje izdvojićemo korake 1) 2) i 3) u jedan potprogram i korake 5) 6) i 7) u drugi (program 1)

Ovim rutinama nije potrebno naročito objašnjenje Pre povratka iz potprograma sys_alt se ne omogućuje prekid (nema EI naredbe) jer se on poziva pošto je već nastao interapt Pre početka upotrebe alternativnog seta potrebno je izvršiti CALL usr_alt a kada nam ne bude više bio potreban — CALL sys_alt, pa EI

Postoji niz slabosti ovakve upotrebe alternativnog seta Na primer nemoguće je menjati ROM, a to znači da će se u ćelijama #39 #3A ROM-a uvek nalaziti adresa kernalove rutine za obradu prekida Znači, dok koristimo alternativni set donji ROM bezuslovno mora biti „isključen" (osim naravno kada nastupi prekid) Stoga se moramo potpuno odreći svih usluga OS-a

Postoji i drugi način kojim bismo zaobišli ove teškoće ali on zahteva mnogo više truda Trebalo bi napraviti novu rutinu za obradu prekida koja bi čuvala alternativni set i ispunjavala iste uslove kao i postojeća Možda bi tako nešto ogadilo ručak i ostale obroke čak i najzagriženijim hakerima, pa ćemo to ostaviti za bolja vremena Uz to trebalo bi uraditi još mnogo stvari

## Sa operativnim sistemom

Prilikom svakog poziva operativnog sistema potrebno je izvršiti istu proceduru kao kod interapta To bi moglo ovako da izgleda

| | |
|---|---|
| CALL sys_alt | ,vrati alternativni set tako da može da ga koristi OS |
| CALL sys_rout › | pozovi potrebnu rutinu pre ovoga nije potrebno EI — ta naredba će ionako biti izvršena posle „uključivanja donjeg ROM-a) |
| CALL usr_alt | ,vrati nazad korisnički alternativni set |

Sve ovo izgleda lepo ali će za samo 10 poziva biti protraćeno 60 bajtova Protraćeno jer se svaki poziv može obaviti sa 3 bajta Jedno rešenje bi moglo da bude upotreba jednog slobodnog restarta a tim da adresa (sys_rout) sledi posle instrukcije RST Na takav način je regulisan glavni džamp-blok pa svaki njegov ulaz zauzima 3 bajta Pošto je jedan restart već odvojen u ovu svrhu, zašto bismo trošili sledeći? S druge strane, ova rutina nije ni izdaleka tako komplikovana kao ona za obradu interapta

Opet ćemo, kao u prethodnom rešenju, malo preurediti nultu stranicu RAM-a U glavnom džamp-bloku koristi se RST #08 Tamo je opet neki džamp koji ćemo zameniti skokom na našu rutinu Prethodno moramo zapamtiti staru adresu

Pored glavnog u memoriji „amstrada" se krije još jedan džamp-blok On koristi drugi restart za poziva rutina OS-a — RST #28 (FIRM JUMP) Preko tog džamp-bloka poziva se amstradov kalkulator

U programu 2 pored ova dva restarta uzeti su u obzir i RAM LAM i FAR CALL Tako se dati program može koristiti i sa bejzikom Kada počinjete upotrebu alternativnog seta potrebno je pozvati rutinu start a kada završavate — potprogram end Prvo važi ako rutinu pozivamo iz bejzika a ako to činimo iz mašinskog programa potrebno je izbaciti naznačena linija Rutina end se može koristiti u oba slučaja Za one koji se još sećaju priče o brzini i alternativnom setu da kažemo jedan podatak sa programom 2 bejzik interpreter će biti sporiji za oko 25%, dok će OS raditi istom brzinom To je cena koja mora da se plati ako želimo da alternativni set nađe mesta u našim programima Još nešto — ako ste sigurni da se neka od kernalovih rutina koje su uzete u obzir u ovom programu neće koristiti možete slobodno da izbacite sve što se na nju odnosi S druge strane ako nameravate da koristite neku rutinu koja ne ulazi u ovaj izbor bilo bi korisno da je ubacite To možete da učinite na način analogan datom sagledavajući razliku između programa 1 i 2 Nadamo se da ste zapazili da neke rutine dobijaju podatke preko steka (tzv inline parametera) na šta takođe treba obratiti pažnju

Na kraju naglasimo još jednom da donji ROM ne sme biti „uključen" ni pod kakvim uslovima ako alternativni set nije prethodno prebačen na sistemsku varijantu

Autor ovog teksta se nada da će ovo biti podstrek za pisanje programa koji koriste alternativni registarski set a to znači programa koji neće biti napisani da bi zaboravljani čamili na polici za kasete

***Dejan Muhamedagić***

# sam svoj programer

Za početak je neophodno da se dogovorimo na koji će način program biti razvijen Svako ko želi može napisati deo za koji smatra da je potrebno da se pojavi u celini Verujemo da je ovo ne samo najlakši već i najbolji način jer se možete koncentrisati na manje probleme i daleko ih efikasnije rešavati nego da je ceo posao zadatak jednog autora Zbog toga je neophodno da uvedemo pravila igre kojih se svi moraju pridržavati ako žele da se ono što rade lako uklopi u sistem

Bez obzira koji problem rešavate sigurno je da će se rutine za obavljanje nekih standardnih zadataka ponavljati Da ne bi svi pisali sve očigledno je da se problem mora razbiti u dve osnovne celine

1) proširenje operativnog sistema
2) proširenje programskog jezika
Redosled rešavanja je jasan

## Osam pravila igre

Da bismo demonstrirali način izvođenja počećemo od samog kraja U programu koji će proširiti bejzik bilo kojom naredbom koja se odnosi na disk jedinicu mora se naći deo koji će izvesti proceduru otvaranja datoteka i očitavanja error kanala radi provere da li takva datoteka postoji Ako ste nešto slično već pisali znate da je ovo najdosadniji deo koji se često ponavlja i za koji uvek treba odvojiti bar 60-tak bajtova koda Pogledajte PRIMER 1 koji obavlja isti zadatak pod uslovom da posedujete samo deo proširenje operativnog sistema koje smo napisali i koje dajemo ovom broju (objašnjanja kasnije) Zapravo samo prvih 8 linija programa obavlja zadatak — ostalo je dato da bi se oformila logička celina

Objasnićemo kako smo operativnom sistemu dodali nove funkcije i pravila kojih smo se pridržavali Evo odmah dve pravila tehničke prirode

*1 Sve rutine moraju biti napisane u asembleru i date u obliku izvornog listinga a ne date liste sa rezultujućim kodom*

*2 Program ne sme koristiti slobodnu memoriju i skokove unutar sebe direktnim adresiranjem lokacija već preko labela*

Prvo pravilo omogućava da jednostavno kombinujemo delove ali i, mnogo važnije da adresa na koju program treba asemblirati može biti slobodno menjana U tom smislu je neophodno i drugo pravilo Program možete razvijati na bilo kom mestu u memoriji, ali ako treba da se kombinuje sa radovima drugih autora od vitalnog je značaja da se može premestiti bilo gde Ovo ne predstavlja nikakvo ograničenje U izvornom programu potrebni radni prostor obeležite labelom i adresirajte ga isključivo njenim navođenjem Ako deo nije uokviran samim kodom već se nalazi na kraju obavezno naznačite (komentarom) koliko bajtova je potrebno U prve dve nove rutine

```
PRIMER 1
--------
        LDA  #37        ; FUNKCIJA EXOPEN
        STA  2
        LDA  #(2*16+8)  ; SA I FA
        LDX  #<NAZIV
        LDY  #>NAZIV
        JSR  820        ; POZOVI EXOPEN
        BCC  OK
        LDA  #22        ; NASTALA GRESKA  POZOVI 622 (CLOSE)
        STA  2
        TXA
        JSR  820
;                       ; OSTATAK PROGRAMA ZA OBRADU GRESKE
;
OK                      ; NEMA GRESKE  NASTAVI RAD
;
NAZIV   ASC "FILENAME"
        .BYT 0
```

```
; -------------------------------------------------
; /                                  FUNKCIJA #39 /
; /                  EXOPEN                       /
; /                                               /
; / ULAZ: X,Y   POCETNA ADRESA NAZIVA DATOTEKE    /
; /             ZNAKA KRAJA NAZIVA BAJT 0         /
; /        A - SA*16 + FA    IIIII IIIII          /
; /                          (-SA  -FA-)          /
; /                                               /
; / IZLAZ:  X - LOGICKI BROJ VEZE                 /
; /                                               /
; / SACUVAN SADRZAJ REG : -                       /
; -------------------------------------------------
EXOPEN  PHA             ; SACUVAJ SA,FA
;
; POSTAVLJANJE NAZIVA DATOTEKE
;
        STX  187        ; POCETNA ADRESA NAZIVA
        STY  188
        LDY  #255
NAMEEND INY             ; NADJI DUZINU NAZIVA
        LDA  (187),Y
        BNE  NAMEEND
        STY  183
;
; ODREDJIVANJE LOGICKE ADRESE
;
        LDA  152        ; BROJ OTVORENIH VEZA
        TAX
        BEQ  SETFA
SEARCH  CMP  600,X      ; NADJI NAJVECU LOGICKU ADRESU
        BCS  LESR
        LDA  600,X
LESR    DEX
        BNE  SEARCH
SETFA   CLC
        ADC  #1         ; POVECAJ ZA JEDAN
        STA  184        ; POSTAVI LA
        PLA
        PHA
        AND  #%00001111 ; PRVA ADRESA (FA)
        STA  186
        PLA
        LSR             ; SEKUNDARNA ADRESA
        LSR
        LSR
        LSR
        STA  185
        JSR  65472      ; KERNAL OPEN RUTINA
        BCS  EROPEN
        LDA  186        ; DATOTEKA NA DISKU ?
        CMP  #8
        BCC  EROPEN
        LDY  #40        ; POZIV FUNKCIJE #40 - IDERROR
        STY  2
        JSR  820
EROPEN  LDX  184        ; LOGICKI BROJ POD KOJIM JE
        RTS             ; VEZA OTVORENA

SLIKA 1
```

koje obavljujemo, već smo nekim delovima u nultoj i prvoj strani odredili namenu, pa bi bilo najjednostavnije da ih za to i koristite Pored spiska funkcija i kodova grešaka oformićemo i mapu novih sistemskih promenljivih Ovi spiskovi će jasno biti prošireni pa u dokumentaciji uz program obavezno dodajte ono što u njima treba da se nađe

*3 Sve funkcije operativnog sistema (postojeće i nove) pozivaju se navođenjem broja funkcije u memorijskoj lokaciji 2 i uvek istom instrukcijom JSR 820*

```
; -------------------------------------------------
; /                                  FUNKCIJA #40 /
; /                 IDERROR                       /
; /                                               /
; / ULAZ:  A - BROJ DISKA (>8)                    /
; / IZLAZ: A - BROJ GRESKE AKO JE CARRY=1         /
; /        679- TEKST GRESKE SA DISKA             /
; /                                               /
; /                                               /
; / SACUVAN SADRZAJ REG.: -                       /
; -------------------------------------------------
IDERROR CMP  #8
        BCC  LLI
        JSR  65460      ; TALK
        LDA  #%01101111 ; SA=15
        JSR  65430      ; TKSA
        LDY  #0         ; CITANJE DISK STATUSA
INPUTER JSR  65445
        STA  679,Y
        INY
        CMP  #13        ; KRAJ PORUKE ?
        BNE  INPUTER
        JSR  65451      ; UNTLK
        LDA  679        ; PRVE DVE ASCII CIFRE
        AND  #%00001111 ; KONVERTUJ U BIN  BROJ
        ASL
        ASL
        ADC  679
        SEC
        SBC  #48
        ASL
        ADC  680
        ADC  #52        ; I DODAJ 100
        CMP  #100       ; NEMA GRESKE ?
        BEQ  IOOK
        SEC
        BCS  LLI
IOOK    CLC
LLI     RTS

SLIKA 2
```

Time se obezbeđuje potpuna nezavisnost programa od apsolutnih adresa Da bi ovo bilo moguće dopisali smo rutinu koja predstavlja okosnicu sistema Ona je data na slici 3 i nećemo se upuštati u njeno detaljnije objašnjavanje Dovoljno je da kažemo da na osnovu broja funkcije koju pozivate izračunava pozivnu adresu i vrši skok na nju uz očuvanje sadržaja svih registara jer se u njima najčešće nalaze potrebni ulazni parametri Funkcije sa brojem 0—38 su standardne KERNAL rutine i mogu se pozivati i direktno ako je to od značaja (eventualno zbog brzine izvršavanja) ali ovo ne treba zloupotrebljavati

*4 Sve ulazne i izlazne parametre treba prenositi putem registara a izuzetno ih ima više mogu se koristiti i memorijske lokacije*

*5 Funkcije za koje se očekuje da će najčešće biti pozivane više puta za redom verovatno u nekoj petlji treba da sačuvaju originalni sadržaj registara koji su imali pri njenom pozivu*

*6 Ako u toku rada rutina detektuje neku od grešaka pri izlazu CARRY zastavica mora biti setovana a kod greške se mora naći u akumulatoru ako nema greška CARRY zastavica pri izlazu mora biti rese-*

***Objavljivanjem sheme kertridža načinili smo tek u „Računarima 10" prvi korak — sada prelazimo na daleko interesantniju stvar: kreiranje programa koji ćemo u njega smestiti. Iako bi nekoga zadovoljilo da u kertridž smesti neki od programa koji često koristi i tako skrati muke oko njegovog stalnog učitavanja sa spoljne memorije, mi smo izabrali teži, ali zato daleko interesantniji put — udruženo pisanje potpuno novog roma! Krajnji cilj je iskorišćenje svih hardverskih potencijala kojima raspolaže „komodor". Uz to ćemo se truditi da maksimalno povećamo konfor u radu, naročito sa perifernim jedinicama. Nedostatak koji oseća najveći broj korisnika je nemogućnost rada sa grafikom visoke rezolucije. Odgovor je SIMON'S BASIC, ali je taj program potpuno zagušen neverovatnom količinom novih (i nepotrebnih) naredbi i deluje kao lečenje kompleksa jednog računara koji „može mnogo, ali se to ne vidi". Ako vam se dopada rešenje koje vam nudimo, prionimo zajedno na posao.***

```
; INICIJALIZACIJA                                      SLIKA 3

        LDA  #<COMM        ; NA ADRESI 029 CE SE
        LDX  #>COMM        ; NACI NAREDBA
        LDY  #76           ; JMP COMM
        STY  020
        STA  021
        STX  022
        RTS
;-----------------------------------------------------
;/              DISPECER NAREDBI                     /
;-----------------------------------------------------
COMM    PHA                ; SACUVAJ REG  A,X
        PHA
        TXA
        PHA
        LDA  2
        CMP  #41           ; POGRESAN BROJ FUNKCIJE ?
        BCC  OK            ; BROJ OK  IZA SADA DE 40!
;
; ERROR EXIT
;
        PLA
        TAX
        PLA
        PLP
        LDA  #255          ; GRESKA #255   NEPOSTOJECA
ERROR   SEC                ;               FUNKCIJA
        RTS
;
;  ODREDJIVANJE POZIVNE ADRESE
;
OK      CMP  #39           ; FUNKCIJE #0-38 SU STANDARDNE
        BCS  NEWFUN
;
; STANDARDNE FUNKCIJE KERNALA
;
        ASL                ; BROJ FUNKCIJE * 3
        CLC
        ADC  2
        ADC  #<65409       ; + POCETAK KERNAL TABLICE
        STA  023
        LDA  #255          ; VISI BAJT ADRESA U KERNAL-U
        STA  024           ; UVEK 255
;
; EXECUTE
;
EXEC    PLA                ; LOKACIJE 023 024 SADRZE
        TAX                ; ADRESU FUNKCIJE
        PLA
        PLP
        JMP  (023)         ; IZVEDI FUNKCIJU !!
;
; NOVE FUNKCIJE OPERATIVNOG SISTEMA
;
NEWFUN  SBC  #39           ; OFSET ZA TABLICU ADRESA
        ASL                ; NOVIH FUNKCIJA
        TAX
        LDA  FTAB,X
        STA  023
        LDA  FTAB+1,X
        STA  024
        CLC
        BCC  EXEC
;
; TABLICA ADRESA NOVIH FUNKCIJA
;
FTAB    BYT <EXOPEN, >EXOPEN
        BYT <IOERROR, >IOERROR
```

tovara. *Nije dozvoljeno nedefinisano stanje niti drugačija signalizacija.*

Ovakav način je već postao standard za prijavljivanje grešaka pa smo ga jednostavno preuzeli iz postojećeg sistema. Da bi na neki način čvršće povezali sa računarom i operativni sistem disk jedinice preuzeli smo i njegove brojeve grešaka ali ih uvećavamo za 100. Tako pri izlazu iz novih rutine operativnog sistema neće biti pravljena razlika u načinu prijavljivanja grešaka računara i diska. Ako se otvara datoteka na disku koji nije uključen KERNAL OPEN rutina će javiti grešku br. 5 i postaviti carry zastavicu, pa je dovoljno da vaše rutine prekine rad i samo prosledi ovo stanje na izlaz. Ako pak taj deo prođe u redu ali se, recimo, u disk jedinici ne nalazi disketa, ovo se mora posebno ispitati očitavanjem stanja preko sekundarne adrese 15 kada će disk poslati poruku (ne zaboravite format je ASCII a ne binarni) broj 74 (Drive Not Ready).

Korisniku vaše rutine treba da bude svejedno gde je nastao problem. U akumulatoru se u oba slučaja mora naći kod greške u prvom slučaju 5, u drugom 174, što vi treba da obezbedite. Jednu takvu rutinu smo razvili IOERROR na slici 2, kao primer kako treba da izgleda prilog, pa se ona može koristiti za ovu namenu. Pošto operativni sistem koristi kodove grešaka od 0 do 9 i 240 a za disk su namenjene one preko 100, ostaje nam prostor od 10 do 99. Vi treba samo da navedete tekst greške a mi ćemo joj dodeliti broj i uvrstiti je u mapu da bismo izbegli preklapanja.

BROJEVI FUNKCIJA OPERATIVNOG SISTEMA:

| | | | |
|---|---|---|---|
| 0 – | CINT | 21 – | OPEN |
| 1 – | IOINIT | 22 – | CLOSE |
| 2 – | RAMTAS | 23 – | CHKIN |
| 3 – | RESTORE | 24 – | CHKOUT |
| 4 – | VECTOR | 25 – | CLRCHN |
| 5 – | SETMSG | 26 – | CHRIN |
| 6 – | SECOND | 27 – | CHROUT |
| 7 – | TKSA | 28 – | LOAD |
| 8 – | MEMTOP | 29 – | SAVE |
| 9 – | MEMBOT | 30 – | SETTIM |
| 10 – | SCNKEY | 31 – | RDTIM |
| 11 – | SETTMO | 32 – | STOP |
| 12 – | ACPTR | 33 – | GETIN |
| 13 – | CIOUT | 34 – | CLALL |
| 14 – | UNTLK | 35 – | UDTIM |
| 15 – | UNLSN | 36 – | SCREEN |
| 16 – | LISTEN | 37 – | PLOT |
| 17 – | TALK | 38 – | IOBASE |
| 18 – | READST | 39 – | EXOPEN |
| 19 – | SETLFS | 40 – | IOERROR |
| 20 – | SETNAM | | |

SPISAK KODOVA GREŠAKA:

0–9, 240 (vidi dodatak "Računari 6", str. 22)
100 – 174 (prema uputstvu za VC1541)

*7. Rutine koje pripadaju operativnom sistemu ne smeju se oslanjati na bejzik ROM.*

I na kraju pravilo 8.

*8. Dokumentacija uz svaku novu rutinu treba da sadrži:*
*— simbolički naziv,*
*— kratak opis namene i načina rada,*
*— potrebne ulazne parametre i način njihovog prenošenja u rutinu,*
*— izlazne parametre i način prenošenja (sem ako nastane greška što smo standardizovali)*
*— spisak lokacija u prve četiri strane memorije (0—1023) koje se koriste,*
*— koji su sadržaji registara očuvani,*
*— dužinu programa u bajtovima i potrebnu količinu radne memorije.*

## Otvaranje datoteke

Poštujući sve ove propozicije, opisaćemo postupak koji treba primenjivati u radu na primeru rutine EXOPEN (Extended Open) koju smo razvili. Koristićemo Profi Ase 64 (koji smo prilagodili za kartridž redakcija vrši programiranje eproma). Prvo treba prekucati program sa slike 3 koji je pri razvoju programa neophodan jer se njime obezbeđuje standardizovan način pozivanja funkcija. Primećujete da nema ORG adresa. Upotrebite bilo koju — za početak će biti pogodna i 49152 (C000). Zatim prekucajte listing sa slike 2 koji predstavlja novu funkciju (broj #40 IOERROR status disk jedinica) koje će nam biti potrebne za ostvarivanje EXOPEN. Isto ćete uraditi sa svim ostalim rutinama drugih autora koje budemo objavljivali (ako su potrebne za realizaciju vaše) i u tablici FTAB dodajte naziv vaše rutine. Time će one samo privremeno dobiti broj i to prema mestu u tablici (prva rutina u tablici ima broj 39) dok ćemo joj konačan broj dodeliti mi kada je definitivno uvrstimo u spisak. Sve ovo može izazvati malu zabunu — kako da razvijate program kada ne znate tačne brojeve svih postojećih funkcija? Zato je neophodno da se pridržavate sledećeg: brojevi funkcija koje su već objavljene moraju se poštovati pa od ovog trenutka važi da EXOPEN nosi broj 39 a IOERROR 40 i njihovi nazivi se moraju naleziti u FTAB u redosledu koji smo dali. Ovo važi čak i ako ih ne prekucate pri razvoju vaše, u tom slučaju negde na kraju, da asembler ne bi prijavljivao grešku samo dodajte dve labele sa ovim nazivima i recimo NOP instrukcijom. Kada završite razvoj listing vaših rutina izdvojite i samo njega pošaljite. Ako ih ima više i jedna poziva drugu onda brojevi funkcija takođe treba da budu dati simbolički. Da bi pozivanje funkcija preko broja bilo moguće potrebno je da inicijalizujete program. Ovo obavljaju prvih 6 instrukcija sa slike 3 pa ih treba izvesti pre startovanja vašeg programa.

EXOPEN je zamišljena kao jednostavnija rutina za otvaranje datoteka prvenstveno na disku. Pošto za ovu operaciju treba pozvati 3 KERNAL rutine (SETNAM, SETLFS i OPEN) sa ukupno 6 bajtova ulaznih parametara, a postupak se često pojavljuje u programima, smatrali smo da je potrebno znatno smanjiti broj potrebnih instrukcija za ovaj zadatak. Tako se u X i Y registru nalazi adresa početka naziva datoteke, dok smo navođenje dužine izbegli tako što naziv mora završavati bajtom 0. Sekundarna adresa i broj uređaja se mogu sasvim fino spakovati u jedan bajt (bitovi 0—3 = broj uređaja, 4—7 = sekundarna adresa), a EXOPEN će sam odrediti logički broj pod kojim je veza otvorena i ako je otvaranje proteklo u redu vratiti ga kao izlazni rezultat u X registru.

NASTAVAK NA 55. STRANI

# Matematički softver To može i bolje (4)

# logaritamska funkcija

Izrada algoritma za izračunavanje vrednosti logaritamske funkcije na računaru sastoji se od više faza. Prva faza je utvrđivanje opsega argumenata za koji se može računati logaritam. Druga faza je utvrđivanje osnovnog intervala u koji će se preslikavati svi argumenti. Treća je izbor aproksimacije u osnovnom intervalu. Četvrta faza je identifikovanje kritičnog intervala u kome je izračunavanje vrednosti algoritama posebno delikatno zbog moguće velike relativne greške koju naravno treba sprečiti. Peta faza je pisanje programa. Dakle, za programera ima više problema koje treba uspešno rešavati i rešiti.

Korisnik računara veruje da je dovoljno za zadati argument X pozvati firminu funkciju LOG i ona će izračunati vrednost LOG(X) u koju se (korisnik računara) može pouzdati. Ta greška može da ima (i obično ima) proizvoljan broj netačnih bitova mantise za programerski odabranu (neprijatnu) vrednost argumenta. Koja su to vrednosti argumenta korisnik računara ne sluti. Ako uspe samostalno da ih otkrije, to znači da ima sklonosti ka programiranju. Razvijanju tih sklonosti posvećena je ova serija tekstova.

## Na osnovni interval

Vrednost logaritamske funkcije može se izračunati za sva pozitivna vrednosti argumenta. Ovde imamo u vidu pre svega računare komodor 64, spektrum i šarp MZ 700/800, koji imaju odvojen 1 bajt za smeštaj karakteristike k i 4 bajta za smeštaj mantise m. Broj X u pomenutim računarima predstavljen je se

$$X = \pm 2^k m$$

sem nula koja je predstavljena nulama.

Karakteristika k je ceo broj za pomenute računare od −127 do 127:

$$K = -127\ (1)\ 127$$

pa je opseg argumenta $2^{-128} \leq |X| < 2^{127}$ ili grubo od $10^{-38}$ do $10^{38}$.

Mantise m uzima vrednosti od 1/2 do $1-2^{-J}$ u koraku po $2^{-J}$

$$m = 1/2\ (2^{-J})\ 1-2^{-J}$$

gde J označava broj bitova mantise (za pomenute računare J=32 pa je tačnost $2^{-32}$ skoro 10 značajnih cifara). U prvom bajtu broja nalazi se karakteristika uvećana za 128. Umesto vodećeg bita normalizovana mantise (koji je uvek 1) zabeležen je znak broja (1 za negativne, inače 0).

Izrada algoritma i programa za logaritamsku funkciju je nezamisliva bez svođenja argumenta na osnovni interval. Razlog za to je nemogućnost primene nekog potencijalnog, verižnog ili kakvog dugog razvoja za proizvoljno pozitivno X. Dakle, ne radi se o tome da li svesti argument na osnovni interval, već kako ga svesti.

Način predstavljanja brojeva u tome pruža nesumnjivu pomoć. Naime, kako je za pozitivno X

$$X = 2^k m$$

onda je

$$\log(X) = k \log(2) + \log(m)$$

log(2) je poznata konstanta pa je ne treba iznova računati. Računanje log(X) proizvoljnog argumenta X svodi se na izračunavanje logaritma mantise, jedno sabiranje i jedno množenje. Izračunavanje logaritma mantise je znatno jednostavnije jer se vrednost mantise kreće od 1/2 do 1. Umesto intervala [1/2, 1) uzimaju interval [1, 2) a neki uzimaju (sqr(1/2), sqr(2)). Uopšte može se predstavljanje

$$X = 2^k m$$

zameniti se

$$X = 2^l f$$

gde je

$$l = K-1 \quad F = 2m \text{ za } 1/2 < m \leq g$$

$$l = k \quad f = m \text{ za } g \leq m < 1$$

Tako se dobija za očuvanje tačnosti izračunavanja pogodniji oblik

$$\log(X) = l \log(2) + \log(f)$$

gde je $g \leq l < 2g$ u $1/2 \leq g \leq 1$.

Krajnje vrednosti za g su vrlo nepovoljne za računanje. Pojedini metodi su najuspešniji za g=sqr (1/2), ali to nije opšte pravilo.

Prema tome izračunavanja vrednosti logaritma proizvoljnog argumenta jednostavno se svodi na izračunavanje vrednosti logaritma broja između g i 2g. To „jednostavno se svodi" je ponekad čist optimizam. Naime, ne želimo da pravimo reklamu jednom (domaćem) proizvođaču računara, ali taj računar sadrži program koji je suštа negacija fraze „jednostavno se svodi". Ako je argument veći od gornje granice (2g) program polovi argument i dodaje međurezultatu log(2), zatim ponovo itd. Ako je argument manji od donje granice (g) program udvostručava argument i oduzima log(2) od međurezultata, zatim ponovo itd. Broj tih ponavljanja i dodavanja (ili udvostručavanja i oduzimanja) je povaći čak i za tako skroman opseg brojeva $2^{-128} \leq x < 2^{127}$. Proizvođač računara pitanje o trajanju izračunavanja po ovom algoritmu smatra nepristojnim. Pohvalno je što su umesto preuzimanja tuđe pameti pokušali sami da nešto urade. Ipak izrada matematičkog softvera proizvoda koji ide u prodaju ne sme se prepustiti početnicima.

Svođenje na osnovni interval je zaista jednostavno ako se karakteristika uzme iz prvog bajta argumenta X. Računar SHARP MZ700 u HuBASICu ima funkciju

VARPTR(X)

koja daje adresu prvog bajta argumenta X. Kod spektruma i komodora to se postiže na komplikovaniji način jer se mora promenljiva čiju adresu tražimo nalaziti na odgovarajućem mestu u programu.

Karakteristika k je sadržaj prvog bajta promenljive manje 128. Ako se u prvi bajt upiše broj 128, onda se na adresi promenljive X nalazi m. Poređenje nove promenljive m sa g omogućava da se sadržaj 128 eventualno zameni sa 129 uz smanjenje upamćene karakteristike k za 1. Dakle testiranje na g treba da se obavlja samo jednom (a ne stotinu puta).

Kako izračunati log(X) u intervalu od g do 2g?

## Obično se preporučuje

Jedan od načina je po definiciji

$$\log(x) = \int_1^x dx/x \qquad (x>0)$$

To nije tako beznadežan metod ako se integral računa na valjan način. U ovoj seriji napisa o matematičkom softveru još smo daleko od integracije jer treba najpre umeti izračunati vrednosti matematičkih funkcija, pa tek onda našto njima računati. Šta je valjan metod integracije je pitanje na koje će se odgovoriti posle pregleda izračunavanja matematičkih funkcija.

Neki autori za računanje vrednosti logaritma preporučuju formulu (1)

$$\log(x) = \lim_{h \to 0} (x^h - 1)$$

Preporučuju, jer je sami ne koriste i to zato što se ni ne može koristiti na računaru. Na slici 1 za nekoliko vrednosti argumenta x prikazana je

```
[illegible]
[illegible]
3 [illegible] GOSUB 4 [illegible]
4 [illegible] I<0 STOP
5 [illegible]*256+PEEK(4[illegible])
6 [illegible] 128  POKE I,128
7 [illegible] 067812 K=K-1  POKE [illegible]
8 [illegible])/(T+1)  T=S*S
9 [illegible]*T+.142857142B6 )*
  T+ [illegible]3333333333 )*Y*S+S)*2
  +K*[illegible]18056 +K*.6875  RETURN

1 REM  LO[illegible]  ZX SPECTRUM
2 LET X=0  LET I=0
3 INPUT X  GOSUB 4  PRINT Y  GO TO 3
4 LET Y=X  IF Y<=0 THEN STOP
5 LET I=PEEK(23[illegible]7B)*256+PEEK(23627)+7
6 LET K=PEEK( ) 128  POKE I,128
7 IF Y [illegible]7071067812 THEN LET K=K-1
  POKE [illegible]
8 LET S=(Y-[illegible]) (Y+1) LET Y=S*S
9 LET Y=(((.[illegible]368 *Y+.142857142B6 )*
  Y+[illegible])*Y+.3333333333 )*Y*S+S)*2
  +K*.00564718056 +K*.6875  RETURN

1     LOG(X)  SHARP MZ-700 (HuBASIC)
3 INPUT X  GOSUB 4  PRINT Y  GOTO 3
4 Y=X  IF Y<=0 STOP
5 I=VARPTR(Y)
6 K=PEEK(I)-128  POKE I,128
7 IF Y<.7071067812  K=K-1  POKE I,129
8 S=(Y-.75-.25)/(Y+1)  Y=S*S
9 Y=((((.113685*Y+.142857142B6 )*
  Y+.2)*Y+.3333333333 )*Y*S+S)*2
  +K*.00564718056 +K*.6875  RETURN
```

relativna greška r u funkciji od h. Inače stepenovanje $x^h$ može se izvršiti se nekoliko uzastopnih korenovanja. Slika pokazuje da se za najpovoljniju vrednost parametra h (približno $2^{-b/2}$, koren iz relativne greške predstavljanja brojeva) dobija samo polovina broja tačnih cifara rezultata. Analitički gledano, sa smanjenjem h trebalo bi da se tačnost formule povećava. Međutim, pored analitičke greške oblika O(h) ovde postoji još i numerička greška oblika O(1/h). Zbir te dve greške je veliki za svako h, pa je ovaj metod neupotrebljiv. Treba uočiti da greška ne potiče od tačnosti izračunavanja člana $x^h$, pretpostavlja se da je taj član izračunat tačno do poslednjeg bita mantise. Greška potiče od razlike $x^h - 1$.

Na slici 2 data je relativna greška formule

$$(2) \qquad \log(x) \approx \sum_{k=1}^{n} (1-1/x)^k/k$$

u funkciji argumenta x. Kako smo usvojili tačnost $2^{-32}$ za g=sqr(1/2) izlazi da je broj sabiraka n=22. Za g=0.53 broj sabiraka je n=19.

Na slici 3 predstavljena je relativna greška potencijalnog razvoja

$$(3) \qquad \log(x) \approx -\sum_{k=1}^{n} (1-x)^k/k$$

u funkciji argumenta x. Za g=sqr(1/2) je n=22, inače za g=0.67 je n=19.

Na slici 4 data je relativna greška formule

$$\log(x) \approx \left( x-1+\sum_{k=2}^{n} (1-x)^k / k(k+1) \right) / x$$

***Korisnik računara zna da je računar snabdeven programom LOG koji za zadanu vrednost argumenta X izračunava približnu vrednost logaritamske funkcije LOG(X). Programer zna za koje argumente X je taj firmin rezultat neupotrebljiv zbog ogromne relativne greške. Programer zna i algoritme tačne do poslednjeg bita mantise rezultata. Prof. dr Dušan Slavić nastavlja seriju „To može i bolje" detaljima iz svog dugogodišnjeg rada u oblasti računarskih algoritama.***

Slika 2. Relativna greška formule $\log(x) \approx \sum_{k=1}^{n} (1-1/x)^k/k$ u funkciji argumenta x.

Slika 1. Relativna greška formule $\log(x) \approx (x^h-1)/h$ u funkciji argumenta h.

Slika 3. Relativna greška formule $\log(x) \approx -\sum_{k=1}^{n} (1-x)^k/k$ u funkciji argumenta x.

Slika 4. Relativna greška formule $\log(x) \approx (x-1+\sum_{k=2}^{n} (1-x)^k/k(k+1))/x$ u funkciji argumenta x.

Slika 5. Relativna greška formule $\log(x) \approx \frac{t}{1+}\frac{1^2t}{2+}\frac{1^2t}{3+}\frac{2^2t}{4+}\frac{2^2t}{5+}\frac{3^2t}{6+}$ u funkciji argumenta $x = 1+t$.

Slika 6. Relativna greška formule $\log(x) \approx \sum_{k=1}^{n} \frac{2}{2k-1}\left(\frac{x-1}{x+1}\right)^{2k-1}$ u funkciji argumenta x.

Slika 7. Relativna greška koja nastaje zbog nebrižljivog oduzimanja 1 od x u kritičnom intervalu

u funkciji argumenta x. Za g=sqr(1/2) je n=16 inače za g=0.67 je n=16.

Navedene formule očigledno zaostaju za sledećim.

Na slici 5 predstavljena je relativna greška verižnog razvoja

$$(5)\quad \log(x) \approx \frac{t}{1+}\frac{1^2t}{2+}\frac{1^2t}{3+}\frac{2^2t}{4+}\frac{2^2t}{5+}\frac{3^2t}{6+}\frac{4^2t}{7+}\frac{4^2t}{6+} \quad (t=x-1)$$

u funkciji argumenta x. Za g=sqr(1/2) dobija se da je red verižnog razvoja 10.

## ... a trebalo bi

Po efikasnosti posebno se izdvaja sledeća formula

$$\log(x)=2\ \mathrm{arth}((x-1)/(x+1))$$

odnosno odgovarajući razvoj

$$(6)\quad \log(x) \approx \sum_{k=1}^{n} \frac{2}{2n-1}\left(\frac{x-1}{x+1}\right)^{2k-1}$$

Na slici 6 data je relativna greška te formule u funkciji argumenta x. Za tačnost $2^{-32}$ u intervalu x od sqr(1/2) do sqr(2) dovoljno je uzeti n=6.

Ekonomizacija potencijalnih (i drugih razvoja) je posebna tema. Kratko se može reći da se ovaj mali broj sabiraka (n=6) može smanjiti uz istu tačnost u zadatom intervalu ako se izmene vrednosti koeficijenata.

Zaključak treće faze je jasan. Treba koristiti formulu (6) ili odgovarajuću ekonomiziranu formulu.

Identifikovanje kritičnog intervala je lako ako za trenutak zamislimo da je argument x zadan sa nekom relativnom greškom p. Kako je

$$(7)\quad \frac{\log(x(1+p))-\log(x)}{\log(x)} = \frac{\log(1+p)}{\log(x)} \approx \frac{p}{\log(x)}$$

zaključujemo da se za ma kako malu vrednost p u okolini tačke x=1 teorijski relativna greška može proizvoljno velika. To je zato što log(x) ima nulu u tački x=1.

Okolina tačke x=1 je kritičan interval. U kritičnom intervalu vrednost funkcije mora posebno biti brižljivo računata da bi bila vrednost logaritma izračunata sa pristojnom tačnošću. Kada su u pitanju tako često korišćeni programi onda je za programera pristojna tačnost tačnost do poslednjeg bita mantise rezultata.

Ovde se naravno podrazumeva da je argument x ne neki željeni (obično decimalni) broj već binarni sadržaj ćelije x. Uz tu pretpostavku može se dobiti rezultat kod koga je samo poslednji bit mantise neizvestan. To je mogućno ako se razlika x−1 računa na suptilniji način, npr.

x−0.75−0.25

Objašnjenje tog na izgled nepotrebno komplikovanog oduzimanja jedinice leži u predstavljanju brojeva u računaru. Ako je argument x između g=sqr(1/2) i 1 on ima karakteristiku 0, a broj 1 ima karakteristiku 1 $(1=0.5\ 2^1)$.

Oduzimanje x−1 obično nije tako valjano izvedeno (što se može) da se ne izgubi poslednji bit mantise broja x. Videli smo da je funkcija log(x) vrlo osetljiva na promenu argumenta u okolini tačke x=1, formula (7). A ovo je promena argumenta i to potpuno nepotrebno izazvana nebrižljivim oduzimanjem jedinice. Ako se (umesto 1) oduzme 0.75 ne čini se nikakva greška jer brojevi x i 0.75 imaju istu karakteristiku. Zatim se oduzme i (¼), što takođe ne unosi grešku. Ako je x između 1 i sqr(2) nije neophodno ali ne smeta da se x−1 realizuje sa x−0.75−0.25.

Šta se događa kada se x−1 nebrižljivo izračuna prikazano je na slici 7. Kada x teži 1 odozdo relativna greška rezultata raste tako da broj tačnih bitova od J opada na 0.

Ako je x između sqr(1/2) i sqr(2) onda je log(x)=log(f) jer je l=0 pa je član l log(2) takođe 0. Modul člana log(f) je u svim drugim slučajevima mali u odnosu na član l log(2).

Ako l nije 0 onda je član l log(2) pretežan. Veći deo tog člana je

0.6875

mogućno je egzaktno izračunati a ostatak je

l (log(2)−0.6875)

srazmerno mali. Optimalnu tačnost u izračunavanju vrednosti logaritamske funkcije mogućno je postići ako vrednosti log(f) dodamo l (log(2)−0.6875) pa tek ovom zbiru glavni deo rezultata (koji je tačan) 0.6875 l.

U priloženim programima izvršena je ekonomizacija razvoja (6) tako da je n=5.

**Dušan Slavić**



UDRUŽENI PROGRAMERI

# jezički procesori u 16 lekcija

**Lekcija 1:** Lekcije 2 do 14 daju početna znanja koja su neophodna, ali često nedovoljna za trgovinu i korišćenja jezičkih procesora (kada ga već kupite možda ne bi bilo loše da možete da ga koristite i u druge svrhe sem da izazivate zavist kod suseda). Za svežije informacije, pogotovo pri konkretnoj kupovini, sledite Brankin primer. Ako ne znate ko je Branka i šta radi strpite se do lekcije 9.

**Lekcija 2**: U lekcijama 4 do 14 objašnjeno je kako možete da odaberete i kupite odgovarajući jezički procesor za vaš personalac.

**Lekcija 3**: Za preprodaju i ostala piratstva nema odgovarajućih lekcija, izuzev lekcije 16 bar ne u „Računarima“ već upotrebite ličnu maštu i kreativnost ako vi to tako nazivate.

**Lekcija 4**: Kada kupujete jezički procesor imajte na umu da način prevođenja bitno utiče na brzinu izvršavanja programa, veličinu potrebne memorije, udobnost i brzinu prevođenja. Ako ste jedan od onih koji žele da uvek drže konce u svojim rukama, kupite i program koji vam daje mogućnost da kontrolišete efikasnost računara. Ovakvi programi obezbeđuju i opcije kao što su kontrole nad tipovima promenljivih koje se koriste (bit, logička ili celobrojne promenljive) ili kontrola preciznosti. Neki kompajleri daju mogućnost da im se specificira i koji će se memorijski modul koristiti tako da generišu efikasnije programe kada se zna da će kôd biti ograničene veličine.

**Lekcija 5**: Pri izboru jezičkog procesora ne treba zaboraviti da pored potrebne memorije za kompajler i linker treba obezbediti i mesto na disketi ili disku za datoteku sa izvornim programom, kao i datoteke koje se generišu pri prevođenju i linkovanju.

**Lekcija 6**: Standardi za jezike se primenjuju strožije kad su u pitanju veliki računari nego personalci. Ipak i za personalce, posebno one namenjene prvenstveno profesionalcima, postoje određeni standardi. To su najčešće podskupovi ili varijante jezičkih procesora za velike računare.

**Lekcija 7**: Mnogi jezički procesori imaju proširenja za pristup operativnom sistemu

Aristotelian Logicians
2631 East Pinchot Avenue
Phoenix, AZ 85016
USA
(602) 956-7678

BD Software
Box 9
Brighton, MA 02135
USA
(617) 782-0836

Per Brinch-Hansen
Computer Science Department
University of Southern California
Los Angeles, CA 90007
USA

Borland International
4807 Scotts Valley Drive
Scotts Valley, CA 95066
USA
(408) 438-8400

c-systems
Box 3253
Fullerton, CA 92634
USA
(714) 637-5362

C Ware
Box 710097
San Jose, CA 95171-0097
USA
(408) 736-6905

The Code Works
5266 Hollister, Suite 224
Santa Barbara, CA 93111
USA
(805) 683-1585

Computer Innovations
10 Mechanic Street
Red Bank, NJ 07701
USA
(201) 530-0995

Digital Research
Box 579
Pacific Grove, CA 93950
USA
(408) 649-3896

Ellis Computing
3917 Noriega Street
San Francisco, CA 94122
USA
(415) 753-0186

Harvard Associates
260 Beacon Street
Somerville, MA 02143
USA
(617) 492-0660

IBM Corporation
Box 1328
Boca Raton, FL 33432
USA
(800) 447-4700

Integral Quality
6265 20th Avenue
Seattle, WA 98115
USA
(206) 527-2918
(206) 527-2918

Kroll Software
1320 Stony Brook Road
Stony Brook NY 11790
USA
(516) 751-5139

Laboratory Microsystems
4147 Beethoven Street
Mar Vista, CA 90066
USA
(213) 306-7412

Lifeboat Associates
1651 Third Avenue
New York, NY 10028
USA
(212) 860-0300

The Lisp Company
Box 437
Redwood Estate, CA 95014
USA
(408) 354-3668

Logitech
165 University Avenue
Palo Alto, CA 94301
USA
(415) 326-3885

Microsoft
10700 Northrup Way
Bellevue, WA 98004
USA
(206) 828-8080

Miller Microcomputer Service
61 Lakeshore Road
Natick, MA 01760
USA
(617) 653-6136

Modula-2 Research Institute
950 North University Avenue
Provo, UT 84604
USA
(801) 375-7402

Morell Data Systems
3400 Wilshire Blvd, Box 70127
Los Angeles, CA 90010
USA
(213) 257-2026

No Visible Support Software
2400 Center Street Box 1344
Berkeley, CA 94701
USA

O. O. Software (James Gaz)
Box 1512
Madison, WI 53701
USA
(608) 244-6436

Ryan-McFarland Corp.
9957 Soquel Drive
Aptos, CA 95003
USA
(408) 662-2522

Silicon Valley Software
10340 Phar Lap Drive
Cupertino, CA 95104
USA
(408) 725-8890
(408) 725-8890

David Smith
44 Old Musket Lane
Danbury, CT 06810
USA

Snobol4
Box 6995
Rochester, MN 55903
USA

Softronics
36 Homestead Lane
Roosevelt, NJ 08555
USA
(609) 448-2605

The Software Toolworks
15233 Ventura Blvd. Suite 110
Sherman Oaks, CA 91403
USA
(213) 986-4885

SofTech Microsystems
16885 Bernardo Drive
San Diego, CA 92127
USA
(619) 451-1230

Springer-Verlag New York Inc
Computer Science Editor
175 Fifth Avenue
New York, NY 10010
USA

STSC Inc.
2115 East Jefferson Street
Rockville, MD 20784
USA
(301) 984-5400

TeleSoft
10639 Roselle Street
San Diego CA 92121
USA
(619) 457-2700

Unified Systems Software
Box 2644
New Carrollton, MD 20784
USA
(301) 552-0594

Volition Station
Box 1236
Del Mar, CA 92014
USA
(619) 481-2286

Richard Walters
University of California
Davis, CA 95616
USA
(916) 752-2793

Watsoft Products, Inc
158 University Avenue W
Waterloo, Ontario
N2L 3E9 CANADA
(519) 886-3700

Western Digital
2455 McCabe Way
Irvine CA 92714
USA
(714) 557-3550
(714) 557-3550

Na primar možata da proverita veličinu vaša datoteka ili veličinu slobodna mamorije na disku i da na njamu obrišele primanujata ili kralreta datoteku

**Lakcija 8** Određani jezički procesori na mogu da na svim (a Marfijev zakon lako može da učini da to buda upravo vaš) računarima da adresiraju svu mamoriju. Slični problami sa pojavljuju i sa stekom i područjem podataka

**Lakcija 9** Važan taktor pri kupovini jezičkog procesora su i preteča bibliotaka i sredstva za razvoj programa i ispravljanja grešaka

**Lakcija 10** Koji jazici su vam na raspolaganju? Ada APL bejzik C kobol tort tortran lisp logo modula-2 paskal PL/I i prolog

```
                                   * Brenka Dokterovic *
                                   * Jeonyska 231 *
                                   * 23000 Kikinda *
                                   * Yugoslavia *

* Snobol4 *
* Box 6995 *
* Rochester NM 14811 *
* USA *
                                   * February 20, 1986 *

Dear Sir

I am very much interested in obtaining the * Snobol * language
processor for my * Commodore 64 *  Please send me at the above
address the price list for each processor and all the information
concerning the payment  If you have a catalog I would appreciate
 if you could send me one

                                   Sincerely

                                   * Brenka Doktorovic *

Deo teksta između zvezdica zamenite tako da odgovara vašim
željama i potrebama
```

Nekoliko kompanija nudi podskupove ade ali jedino Wastarn Digital nudi potpunu adu — na žalost jedino za svoj hardver. Ako vas podkupovi zadovoljavaju, pišita firmama R R Softwara ili TalaSoft

APL nija našao mnogo pokionika među korisnicima personalnih računare ali ako vi mislita da ja to ono pravo IBM STSC Inc i Softronics vam nuda interpretatore za 8-bitna i 16-bitna računare

Bejzik je još uvak jezik № 1 personalaca i možete ga nabaviti kod Microsofta i Digital Researcha Iako ja bejzik predviđen prvenatvano za mala aplikativna programa tekuća ANSI varzija ja modarnizovala jezik i tako mu proširila polja primanljivosti

Posla bajzika jazik C ja na naprikosnovenom drugom mastu Kompleksna aplikacije koja su za 8-bitna personalna računare pisana isključivo u asembiaru za 16-bitna mašina uglavnom se pišu u C-u Uz C kompajler obično idu i operetivni sistemi inspirisani UNIX-om Za kupovinu pišita firmama BD Software Microsoft Digital Research C Ware ili Computer Innovations

Microsoft Digital Reasearch i Ryan McFarland prodaju kampajlere za kobol

Fort, stek orijentisani jezik uživa fanatičko oduševljenja među korisnicima personalnih računara od prvih dana svoja pojava Koristi sa prvenatveno za pisanja programa za kontrolu ali ja našao i svoju ulogu u obredi taksta kao i programima za poslovna potrabe

Fortran sa na personaicima koristi prvanstveno da bi sa omogućila upotreba valikog softvara koji ja napisan za valika računare Kompajlari su napisani i za 8-bitna i za 16-bitna personalna računare a reailzovana verzija su skoro potpuni podskupovi fortrana 77 između ostalih fortran kompajlare prodaju i Microsoft Digital Research i Silicon Vallay Softwara

Pojava akspertnih sistama približila ja veštačku intaligenciju i korisnicima personalnih računare Jezički procesor lisp vam može pomoći da i vi zakoračata u ovu novu oblast a možata ga kupiti kod Integral Quality Systems Microsoft i The Lisp Company

Među kompjutarskim edukatorima logo uživa napodaljana simpatija Potpuni kompajlari ga čina sva više i aplikativnim jazikom Za potancijalna kupca najbolja ja da sa obrate IBM Digital Research i Krell Software-u

Mnogi modulu-2 smatraju idaalnim prelazom sa paskala na adu i za njih su Springar-Varlag Logitach i Voiltlon Systams obezbedili kompajlare

Virtova dafinicija paskala proširena ja ANSI i ISO standardima koji sa mogu nabaviti kod Microsofta Digital Researcha i SofTacha

Ako imata IBM PC XT/370 možata pušlati 370 PL/1 programe Ako namate IBM PCa a život bez PL/1 vam ja nezamisliv spašća vas Digital Research svojim podskupom G PL/1

Za ona koji na vola lisp, a vola veštačku intaligenciju Prologov britanski interpretar prodaja Springar-Varlag

**Lakcija 11** Sem pomanutih kompanija, jezička procesore prodaju i druga firma i zato ja najbolja da pišeta nakoj od firmi iz tabele za šira obaveštanja

**Lakcija 12·** Ako sta sujeverni ili znate angleski preskočita lekciju 13

**Lekcija 13** Dok na utvrdite svoja znanja angleskog koristite odgovarajuće adaptirano pismo Brenka Doktorović Dolaze teška vremana — na dozvolita da vas pretaknu žana

**Lakcija 14:** Jezički procesori iz lakcija 14 mogu da koštaju i do 2000 dolara pa i više iako sa retko sreću cana veće od 500 dolare Za 50 ili manja dolara Microsoft prodaja bejzik Software Toolwork i The Code Works C Ellis Computing kobol i fortran Arlatotelian Logicians No Vlable Support Software Unified Systama Software prodaju Forth Norell i Software nuda lisp a Modula-2 Research Institute pogodili sta modulu-2! Paskal sa moža kupiti kod Borgland Internationala a kupcima snobola preporučujemo da se pridruže Brenki

**Lakcija 15** Pored firmi i valiki broj pojedinaca prodaja razna specijalna jezička procesore Pored Hansana Dajvids Smita i Ričarda Valtares na tom spisku su i Horovic, Susman i drugi valikani računarstva ali i široj javnosti napoznati pojedinci Spisak njihovih adresa se priprema i [(ako odobri urednik)] Računari' će ga objaviti kad buda gotov Za sad se morate zadovoljiti oficijalnim adresama iz tabele

**Lakcija 16** Pišita nam ako sta autor nakog od jezičkih procesore Sa zadovoljstvom ćemo vas uvrstiti u tabelu autore

***Nada Aleksić***

# Šta ima novo

## Milion po milion — posao

*Čuje se da Sar Klajv Sinklar ima nakih manjih problema sa prikupljanjem desat miliona funti koliko mu ja potrebno za izradu sledeće serija računara koje ja zamislio Zar je moguće da mu finansijeri na veruju?*

## Poljubac za skor

*Dok ovo budata čitali ovogodišnji FEST će već biti završen Neki od vas će na njemu varovatno gledati i amarički film „Povratak u budućnost To ja još jedan iz niza uspešnih filmova koje je finansirao Stiven Spilberg Ono što je za nas interesantno ja da postoji i video igra napravljena po tom filmu U toj igri se po prvi put pojavljuju (ako na računamo honkoška porno igre) poljubaci Da dobro ste pročitali poljubac u video igri! Verovatno ćemo još čuti o toj igri*

## Hit lista

*Da vas obavestimo da sa na top listi Your computer-a na prvom mestu nalazi Elita o kojoj ćate malo detaljnija pročitati u članku Dejana Ristanovića*

## Novosti od Hjusona

*Ako vam se puno sviđeo Paradroid znajte da sa upravo sprema još jedna igra sličnog dizajna i slična zanimljivosti Ime joj je Uridijum i prvi reklamni skrinovi koji su se u štampi pojavili zaista čina da pomislite da se vaš komodor 64 ' pretvorio u pravu arkadnu mašinu Čim se pojava prve kopija dobićete prikaz*

## Brbljivi „mek"

*Ako imata Macintoš obavezno nabavite Smooth talkar — program koji omogućava meku da svaki tekst ispisan na ekranu izgovori jasno i glasno preko svog zvučnika Za samo 99 dolare pod uslovom da sta nabavili varziju 2 0 imate mogućnost da birata muški ili ženski glas Ali avaj niko nija idealan — muški glas govori sa jakim kanadskim akcentom dok ja lepši pol po poreklu iz Škotska Ali ako možete da pređete preko takvih nesavršenosti programa naći ćate ga u svakoj boljoj robnoj kući trafici ili apoteci u Kaliforniji*

*I N*

# Najveći konkursi sveta

## konkurs „Abakus“

Konkurs da ili ne? Pre svega kakav? Najbolji su svakako oni sa nagradama od desetak hiljada dolara ili više ili oni koji donose izuzetnu slavu zar ne? I šta čekaju li Računari ? Što ga već jednom na objave? Dok Računari ne skupa 100 000 dolara za nagradni fond ili na ubede Virta ili Harlana Milsa da učestvuju s vi ih pobedite pogledajmo šta nudi ostatak sveta U sledećim brojevima ćemo s vremena na vreme objavljivati najbolja svetska konkurse u skladu sa sledećim pravilima

1 Konkurs mora biti otvoren još najmanje 6 meseci posle objavljivanja u Računarima

2 Nagrade na konkursu moraju da budu kao u tački 3 ili 4

3 Nagrada mora biti bar 10 000 dolara što više to bolje ili kao u tački 4 Nagrada mora biti takva da pobedniku i svim ostalima nedvosmisleno ukazuje na to da je pobednik bolji programer od nekog dobitnika Tjuringove nagrade bar kad je u pitanju nagradni zadatak

5 O svim pobednicima čitaocima Računara Računari objavljuju reportažu i upoznaju javnost sa našim superprogramerima

6 Posebna nagrada pripada onome ko nas obavesti o nekom konkursu koji potom objave Računari

```
  2 REM "KONKURS CASOPISA ABACUS"
  3 REM "B.A. TEAGUE ZA TRS 80 III"
  5 DIM ARRAY$ (9)
  7 GOSUB 500
  9 DEFDBL B-H,J-Z
 10 N=11111
 20 S=N*N
 30 S$=STR$ (S)
 40 X=LEN (S$)-1
 45 IF X > 9 THEN 300
 47 GOSUB 500
 50 FOR I=1 TO X
 60   C$=MID$(S$,I+1,1)
 65   C=VAL(C$)
 67   ARRAY$(C)="OK"
 70 NEXT I
 80 FOR A = 1 TO 9
 90   IF ARRAY$(A)="BAD" THEN N=N+1: GO TO 20
100 NEXT A
110 LPRINT "N=";N;" S=";S$;"
115 N=N+1:GOTO 20
300 END
499 REM "POTPROGRAM"
500 FOR A=1 TO 9
510     ARRAY$(A)="BAD"
520 NEXT A
530 RETURN
```

7 Sve pravila u vezi sa konkursima menjaju se i proširuju isključivo u skladu i dogovoru sa čitaocima

Ako znate za neki konkurs koji zadovoljava uslove za objavljivanje pišite nam Dok ne dobijamo vaše predloge mi ćemo objaviti nekoliko konkursa koji su nama poznati

Prvi na listi je konkurs Abacus -a izvanrednog lista za popularizaciju računarstva koji izdaje Springer—Verlag (Za one koje interesuje pretplata adresa izdavača je Springer—Verlag New York Inc 175 Fifth Avenue New York NY 10010 USA ) Nagrade na konkursima su skromne pretplata na Abacus i par knjiga sa rešenim zadacima iz računarstva ali su zato redovni učesnici na ovom konkursu recimo Gris i Knut a njih pobediti nije šala! Zadaci su takvi da omogućavaju da konkurs bude stalno otvoren i svako novo rešenje bilo kog zadatka se priznaje pod uslovom da ono ranije nije publikovano na stranicama Abacus -a. Zadaci se numerišu i mi ćemo ih numerisati kao u Abacus -u tako da kad ih rešite možete ih poslati Richard-u V Andree-u University of Oklahoma USA ili uredništvu „Abacus -a

**Problem 1** Odrediti sve pozitivne cele brojeve tako da kvadrat S=N*N sadrži 9 cifara (1 2 3 4 5 6 7 8 i 9) tačno jednom

**Problem 2** Naći sve pozitivne brojeve takve da je suma kubova njihovih cifara jednaka tom broju?

Primer $371 = 3^3 + 7^3 + 1^3$

Ovaj zadatak je rešio Dejvid Gris Red profesor na Kornelu autor divne knjige „Nauka programiranja“) našavši rešenja 1 1634 8208 i 9474 Možete li bolje?

**Problem 3** Znate li šta je palindrom? Na primer Ana voli Milovana ili Udovica baci vodu Unapred i unazad se čita isto Znate li neki engleski? Able was I ere I saw Elba ili Norma is as selfless as I am Ron A da li znate neki kvadrat koji se čita unapred i unazad isto? Sa neparnim brojem cifara

$11 \times 11 = 121$

$26 \times 26 = 676$

$264 \times 264 = 69696$

$22865 \times 22865 = 522808225$

a sa parnim

$836 \times 836 = 698896$

Da li znate neki treći stepen koji je palindrom? Na primer

$7^3 = 343$

$2201^3 = 10662526601$

A za četvrti stepen?

$101^4 = 104060401$

Nađite isto za 5 6 i 7 stepen

Nudimo nekoliko rešenja čitalaca za kvadrat

$798644^2 = 637832238736$

$64030648^2 = 4099923883299904$

I zasad najveći

$6360832858988^2 = 40460119551118611559106404$

Dovoljno?

**Problem 7** Znate li da rešite Hanojsku kulu Imate tri štapa — a b i c — i na štapu a n kolutova tako da je najmanji na vrhu a veći redom ispod do najvećeg koji je na dnu Treba ih premestiti na štap c koristeći štap b kao pomoćni Rešite isti zadatak kad su a) diskovi alternativno crveni i beli a veličinski odnos kao malopre i b) jedan disk se sme prebaciti ili na prazan štap ili na veći disk ali samo ako je disk druge boje. I rešili ste problem 7

***N Aleksić i M. Potkonjak***

# Kolumbovo jaje

## Softverski perpetuum mobile

***Prenos programa sa jedne mašine na drugu predstavlja problem sa kojim se suočava veliki broj ljudi koji se profesionalno ili iz hobija bave računarstvom i laiku je jasno da to ne ide glatko čak ni kada se radi o vrlo sličnim računarima U računarstvu je, međutim, izgleda sve moguće — čak i to da se program za jedan računar izvršava na nekom sasvim drugom.***

Metod koji ovde opisujemo Amerikanci zovu „bootstrapin“ Za one kojima engleski nije jača strana recimo da bootstrap znači šare na čizmama, pa bi bootstraping bio postupak kad čovek uhvati samog sebe za šare od čizama i podigne uvis!

Pretpostavimo da posedujemo mnogo softvera u izvornom kodu (neprevedenom obliku) nekog jezika L i mašinu M1 sa kompajlerom C1 za taj jezik Između ostalog, imamo i izvorni neprevedeni oblik kompajlera C1 pisan takođe u jeziku L

Sors verziju (izvorni kod) nekog programa označavaćemo sa Ps a prevedenu verziju (objektni kod) sa Po

Pretpostavimo da smo nabavili sasvim novu mašinu M2 i tanki smo sa softverom za nju Ako bismo nekako uspeli kompajler za jezik L da prebacimo na računar M2 automatski bi i na M2 imali puno softvera (svaki Po bi preveli na M2 uz pomoć tog kompajlera)

Setimo se da naš kompajler C1 imamo i u sorsu C1s. Zato nam neće biti baš previše teško da ga prepravimo tako da umesto koda mašine M1 generiše kod mašine M2 (naročito ako smo baš mi napisali i kompajler C1s) Čak i ako nam taj posao oduzme prilično vremena to je još uvek jednostavnije nego da za M2 pišemo sve programe u mašincu

Program pisan u jeziku L koji pravi kod za M2 nazovimo C2s Lepo je to što sada imamo C2s, ali kako da ga prevedemo na mašinski jezik mašine M2?

Za to ćemo se poslužiti starim računarom M1 Uz pomoć njegovog kompajlera C1 prevešćemo program C2s

Dobijeni objektni kod C2o može da se izvršava na M1 Uočimo da je C2o program čiji je ulaz neki program pisan u jeziku L a izlaz mašinski kod računara M2

Sada ćemo na M1 pustiti program C2o a kao ulaz podmetnućemo mu baš C2s!!! Dobijeni izlaz C2t je prevod kompajlera C2s na mašinski kod računara M2 tj moguće ga je izvršavati na M2

To je upravo ono što nam je i bilo potrebno — kompajler jezika L na mašini M2

Neverni Toma će reći — ovo je fenomenalna stvar ali samo pod uslovom da u prepravci kompajlera C1s u C2s nismo nigde pogrešili što je malo verovatno To, međutim, i nije neki problem, jer imamo mogućnost jednostavnog testiranja kvaliteta novog kompajlera!

Dokazivanje korektnosti i mnogo trivijalnih programa od jednog kompajlera je mukotrpan posao ali opet ćemo koristiti strašan trik Uz pomoć C2t prevešćemo opet C2s Ako u pisanju C2s nismo omanuli, kao izlaz ćemo dobiti C2t

Jednostavno, zar ne?

***Zoran Obradović***

Programiranje u bejziku

# svirka na „amstradu"

**_Računar „amstrad"/šnajder CPC 464" je opremljen jednim od najboljih zvučnih čipova na tržištu AY-3-8912, koji istovremeno proizvodi tri tona plus šum. Ako se zna da je zvuk sa CPC 464 (kao i na drugim amstrad/šnajderovim računarima) namenjen i za stereo reprodukciju, onda je jasno da je ova mašina prava stvar za kompozitore na kućnom kompjuteru. To nije slučajno, jer je fabrika Amstrad inače poznati proizvođač solidnih audio uređaja._**

Za svaku primjenu zvuka na računalima je vrlo važno na koji način ga možemo ostvariti tj. da li se zvuk ostvaruje pomoću jednostavnih naredbi ovdje u solidnom Locomotive BASIC-u ili tek pomoću strojnih programa. Kod računala CPC 464 ili naprednijih varijanti (664 i 6128) manipulacija zvukom je u velikoj mjeri moguće upravo iz bejzika što predstavlja olakšanje početnicima, ali i naprednijim programerima. Koliko god ta upotreba bila jednostavna, zbog nesnalaženja u osnovnom priručniku zvučne mogućnosti ovog računala mogu ostati neiskorištene pa čak i za korisnike s minimalnim zahtjevima. Tema ovog napisa je kako što bolje iskoristiti zvukovne mogućnosti. Do prave svirke može doći tek nakon sistemskog upoznavanja sa svim osnovnim aspektima upotrebe zvuka za konkretne namjene.

Kod računala CPC 664 se zvuk reproducira preko ugrađenog zvučnika a regulacija glasnoće se vrši pomoću potenciometra s desne strane računala. Ukoliko nas taj zvuk ne zadovoljava (relativno tih, izobličenje) moguće je koristiti ulazno-izlazni (input-Output) port koji omogućuje priključenje na stereo Hi-Fi liniju ili klasično audio pojačalo i tako dobiti jači i kvalitetniji zvuk. U praksi to izgleda da se jedan kanal reproducira preko lijevog izlaza, drugi preko desnog, a treći u stvari daje srednji kanal. Zvučni izlaz je dosta niske razine pa pri radu s nekim pojačalima može biti problema.

## SOUND

Glavna instrukcija za rukovanje generatorom zvuka je instrukcija SOUND koja se može koristiti uz ostale instrukcije tako da daje veće mogućnosti primjena nego kada je koristimo samu za sebe. Instrukcija SOUND je određena sa sedam parametara, od kojih je prvi tzv. status kanala. Status kanala se određuje prema sljedećoj tablici.

### određivanje statusa kanala

| bit | decimalno | značenje |
|---|---|---|
| 0 | 1 | odabira kanal A |
| 1 | 2 | odabira kanal B |
| 2 | 4 | odabire kanal C |
| 3 | 8 | spajanje s kanalom A |
| 4 | 16 | spajanje s kanalom B |
| 5 | 32 | spajanje s kanalom C |
| 6 | 64 | održavanje tona-HOLD |
| 7 | 128 | treperenje tona-flushing |

Dakle prva tri parametra odabiru kanal — za generiranje tona s kanala A odabiramo 1 za prvi parametar ili za zvuk sa sva tri kanala — 7 (1+2+4=7). Ostali parametri će biti kasnije objašnjeni. Sljedeći parametar u instrukciji SOUND određuje visinu tona. U osnovi, generator zvuka koristi klok 4 MHz čiju frekvenciju dijeli sa 16, te daje konačnu frekvenciju od 125 kHz. Ona se zatim dijeli sa brojem — veličinom parametra koji se koristi za visinu tona tako da veći broj daje veći omjer tj. nižu izlaznu frekvenciju. Visina tona koja je zadana kao 4 ili manje daje izlaznu frekvenciju koja je na granici čujnosti ljudskog uha (okvirno 20 Hz—16 kHz) tako da samo oni s dobrim sluhom mogu čuti frekvencije proizvedene s računala koje su definirane vrijednošću parametera 5—8. S druge strane najveća vrijednost parametra visine tona je 4095 što daje rezultirajuću frekvenciju od cca 30 Hz — dosta duboki bas ton. Kod takvih frekvencija se mora suočiti s činjenicom da mali zvučnik ugrađen u računalo ne može reproducirati vrlo niske zvučne frekvencije točno onako kako bismo željeli, odnosno dolazi do manjih ili većih izobličenja.

Trajanje tona određuje treći parametar u odnosu na jednu stotinku sekunde. Na taj način su omogućene varijacije zvučnih efekata kao i muzika koja se može lakše proizvesti. Vrijednost 32767 određuje najduže trajanje koje u krajnjem slučaju može biti 5 minuta.

Četvrtim parametrom se kontrolira jačina zvuka — ako se koristi samo instrukcija SOUND bez instrukcije ENV vrijednost parametra je od 0 koja isključuje generator, do 7 što daje najveću jačinu. Kada se instrukcija ENV koristi uz SOUND raspon jačine se proširuje od 0 do 15. To bi bilo najosnovnije uz instrukciju SOUND jer se momentalno ostale tri parametra mogu zapostaviti tj. pridružiti im vrijednost 0.

Kao primjer se može uzeti instrukcija

```
SOUND 1 250 200 4 0 0, 0
```

koja znači da će kanal A proizvesti visinu tona (određenu s 250) u trajanju od 2 sekunde (200 100) nivo jačine 4. Za potpuniju ilustraciju može se prikazati rutina

```
15 SOUND 2 150 200, 4, 0 0 0
25 SOUND 7 150 200 4 0 0 0
```

koja prvo proizvodi zvuk s kanala B visine tona zadane s 150 u trajanju 2 sekunde a potom to isto proizvodi sa sva tri kanala. Može se vidjeti da je jedina razlika između jednog kanala i sva tri samo u jačini zvuka.

Kao primjer se može izvesti jednostavan generator tona koji reproducira vrijednosti visine tona od 1 do 4095 a to se postiže petljom

```
15 FOR A=1 TO 4095
25 SOUND 1, A 1, 7 0 0 0
35 NEXT A
```

Za bolje razumijevanje generiranja tonova preporučljivo je da početnici koriste neku literaturu iz audio-tehnike.

Na sličan način se zvučni efekti mogu proizvesti rutinom, koja uz modifikacije može imati primjenu pri razvijanju vlastitih programa

```
15 FOR X=20 TO 80
25 SOUND 1 X 1, 7 0, 0, 0
35 NEXT X
```

Ova rutina jednostavno daje visine tonova definirane od 20—80 minimalnog trajanja od jedne stotinke sekunde.

Varijacije jačine proizvedenog zvuka se mogu nazvati i oblikovanjem zvučnih ovojnica-anvelope (engl. envelope). Na primjer, smanjivanje jačine zvuka se može postići rutinom sljedećeg oblika

```
15 A=7
25 FOR B=20 TO 80
35 SOUND 1 B, 1 A 0 0 0
45 A=A-0 1
55 NEXT B
```

Linije 15 postavlje jačinu zvuka na 7, a linija 45 smanjuje tu vrijednost za 0 1 pri svakom koraku petlje Računalo može ponekad zablokirati (crashing) ako parametar jačine zvuka (u ovom slučaju varijabla A) nije cjelobrojna veličine

## MUZIKA

Na amstrad/schneideru je moguće programiranje muzike u trodjelnim harmonijama Međutim oni koji nisu vični komponiranju treba da koriste što jednostavnije muzičke izraze Također im na raspolaganju stoje solidni programi koji olakšavaju komponiranje CPC 464 ima impresivan raspon moguće upotrebljivih nota — točnije 8 oktava Kao što je ranije rečeno broj čujnih visokih frekvencija je relativno ograničen

To praktično znači da nam ne raspolaganju stoji čujan raspon od 5 oktava koji se proširuje ako računalo priključimo na kvalitetniji audio uređaj Računalo ne možemo programirati direktno u muzičkim vrijednostima već moramo notama pridruživati određene vrijednosti za visinu tona što je opisano u VII poglavlju priručnika koji se dobije uz računalo Ostvarivanje jednostavne melodije može se vidjeti na sljedećem primjeru

```
15 A=1
25 WHILE A()0
35 READ A B, C
45 SOUND 1 A B C 0 0 0
55 WEND
85 DATA 239 50 5 213 50 5 190 50
        5 179 50, 5 159 50 5 142
        50 5, 127 50 5 119 100 7
        0 0 0
```

Ovdje se koristi petlja WHILE WEND pomoću koje je melodija kompletirana Željena visine tona trajanje i jačine se dobiju iz podataka u DATA liniji Program je zaključen korištenjem nule za vrijednost visine tona ali treba paziti da trajanja i jačina budu uključeni ili će u protivnom program zablokirati u liniji 35

### Spajanje kanala

Prvi parametar u instrukciji SOUND određuje odabir kanala ali je moguće upravljati i njihovim spajanjem Drugim riječima moguće je odjednom upravljati sa više kanala Za spajanje kanala upotrebljavaju se sljedeće vrijednosti

8 spajanje s kanalom A
16 spajanje s kanalom B
32 spajanje s kanalom C

Tako npr za slanje zvuka na kanal C i spajanje s kanalom A i eventualno održavanje tona postižemo sa 4+8+64=76 što unosimo kao prvi parametar instrukcije SOUND

Kako funkcionira spajanje kanala može se vidjeti iz primjera

```
15 SOUND 12 100 50 5 0 0, 0
25 FOR A=1 TO 1000 NEXT A
35 SOUND 33, 200 50 5 0 0 0
```

Linije 15 je SOUND instrukcije za kanal C koji se spaja s kanalom A (4+8=12) Linije 25 proizvodi malo kašnjenje a linije 35 je SOUND instrukcija za kanal A spojen s kanalom C (1+32 [illegible]) [illegible] kazuje praktičnu upot[illegible]

```
15 SOUND 1, 120 50 5 0 0 0
25 SOUND 2, 50, 100 5 0 0 0
35 SOUND 17, 100 50 5 0 0 0
45 SOUND 10, 150 50 5 0, 0 0
```

Prve dve linije proizvode tonove se kanale A i B ali je trajanje različito za oba kanala Sljedeće dvije linije predstavljaju također instrukcije SOUND za kanale A i B ali nose u sebi brojeve statusa kanala koji prouzrokuju spajanje Prema ovom primjeru moguće je pisati dvodjelne ili trodjelna harmonije Spajanje kanala predstavlje u stvari točnu sinhronizaciju muzičkih dionica

### Održavanje tona

Održavanje tona moguće je postići dodavanjem vrijednosti 64 u vrijednost statusa kanala Primjer

```
15 SOUND 65, 100 100 5 0 0 0
25 SOUND 66 150 100 5 0 0 0
35 SOUND 66 250 100 5 0 0
45 FOR A=1 TO 1000 NEXT A
55 RELEASE 7
```

Instrukcije SOUND u linijama 15 25 i 35 odnose se na sve tri kanala A B i C kojima je dodana vrijednost 64 za postizanje održanja tona Linije 45 proizvodi kašnjenje od otprilike 1 sekunde prije nego što dođe do otpuštanja (RELEASE) instrukcije RELEASE pomiče održavanje na označeni kanal Kod instrukcije RELEASE se odabir kanala postiže kao i prvi parametar u instrukciji SOUND — pridruživanjem određenih vrijednosti statusa kanala Ukoliko je potrebno „otpustiti" više kanala njihovi brojevi statusa kanala se jednostavno zbroje — u prethodnom primjeru RELEASE 7 — (1+2+4)

### „Flushing"

Posljednja karakteristika u instrukciji SOUND koji se može definirati u statusu kanal je flushing Vrijednost koja se upisuje u prvi parametar SOUND instrukcije iznosi 128 Sljedeći primjer ilustrira efekt flushinga

```
15 FOR A=1 TO 50
25 SOUND 130 1 1 0 0 0 0
35 NEXT A
```

Petlje se ponavlja 50 puta a parametar za dužinu trajanja u instrukciji SOUND daje ton u trajanju od 1 sekunde za svaki korak petlje Vrijednost statusa kanala 130 je dobivena iz upotrebe kanala B i flushinga 2+128=130) Jedna od namjena flushinga je i ta da isključuje jedan ili više kanal generatora zvuka što se može vidjeti na primjeru

```
15 SOUND 65, 200, 100 7 0 0 0
25 SOUND 129 1 1 0 0 0 0
35 RELEASE 1
```

Programska linija 15 generira ton trajanja 1 sekunde sa držanjem tona u kanalu A a linija 25 vrši flushing kanala A i ne proizvodi zvuk jer je parametar jačine zvuka jednak nuli Linija 35 otpušta kanal A

## ŠUM

Šum (noise) može biti proizveden i miješan u jednom ili u više tonskih kanala ali istovremeno može biti proizveden samo jedan zvuk šuma U principu nije ni potrebno da se neki ton miješa sa šumom a koristeći malu vrijednost (1 ili 2) visine tona proizvedeni ton može biti nečujen tako da samo šum bude prisutan na izlazu Šum je obično pišteći a visine tona može biti određivane po želji Šum je moguće aktivirati koristeći vrijednosti od 1 do 31 i to kao posljednji parametar instrukcije SOUND Ako se koristi vrijednost 0 šum je potpuno potisnut Vrijednost 31 daje minimalnu visinu tona, a 1 maksimalnu Relativno širok raspon visine tonova šumova je koristan za različite zvučne efekte S obzirom da se šum koristi uglavnom kod igara najčešće se upotrebljavaju oblici koji liče na eksplozije i pucnjeve što prikazuje naredni primjer

```
15 A=12
25 FOR B=7 TO 0 STEP -1
35 SOUND 2 1 20 B 0 0 A
45 A=A+2
55 NEXT B
```

Instrukcije SOUND u liniji 35 proizvodi zvuk na kanalu B ali je ton previsok da bi se čuo Trajanje je jedna petina sekunde pri svakom koraku petlje Jačina i visine tona i šuma se upravlje varijablama A i B Vrijednost varijable B (7) se postupno umanjuje sve do 0 što daje realističnu simulaciju zvuka eksplozija Linija 45 smanjuje visinu šuma dajući na kraju minimalnu visinu Prethodna rutina se može modificirati za npr simulaciju pucnja, samo je potrebno koristiti veći opseg visina šuma te odrediti kraća trajanja zvuka npr

```
15 A=3
25 FOR B=7 TO STEP -3
35 SOUND 1, 1, 3 B 0, 0, A
45 A=A+1
55 NEXT B
```

Primjer rutina za šum može imati i oblik

```
15 FOR X=1 TO 31
25 SOUND 1 0 100 4 0, 0 X
35 NEXT X
```

ili

```
15 FOR X=1 TO 2
25 FOR Y=16 TO 31
35 SOUND 1 0 5+Y, 5, 0, 0 Y
45 NEXT Y
55 FOR Y=31 TO 12 STEP -1
65 SOUND 1 0 7 6 0, 0 Y
75 NEXT Y
85 NEXT X
```

Iz ovih primjera se vidi da u instrukciju SOUND može biti unešen i neki matematički izraz — linija 35

Najraznovrsniji zvučni efekti se mogu postizati varijacijama upotrebe naredbi za generiranje zvuka i njihovih parametara te se na taj način ostvaruje široko polje primjena tonskih mogućnosti računala amstrad/schneider

## ENV

Kod mnogih primjena instrukcije SOUND uz korištenje petlji daje zadovoljavajuće rezultate Međutim tamo gdje je potrebno složenije oblikovanja jačina zvuka ili visine tona potrebno je uz instrukciju SOUND koristiti i instrukcije ENV (envelope — volume/ovojnica-jačine) ili ENT (envelope-tone/ovojnica-ton) Prvi parametar instrukcije ENV je identifikacioni broj (od 1 do 15) koji omogućuje željenu ovojnicu koje će biti pozvana instrukcijom SOUND (njezin peti parametar) Ostale tri broja kod instrukcije ENV određuju broj koraka, po većanje ili smanjivanja jačina zvuka za svaki korak te trajanja (u stotinama sekunde) svakog koraka Broj koraka mora biti u rasponu od 0 do 127 dok koraci s obzirom na jačinu zvuka mogu biti između −128 i +127 Uobičajeno se koristi raspon od −15 do +15 Maksimalno trajanje po koraku je 255 Kao jednostavni primjer za smanjivanja jačina od 15 do 0 u petnaest koraka

trajanja deset stotinki sekunde, može poslužiti instrukcija ENV 1 15 —1 00 Za tvorbu ovojnica se u praksi koristi do 5 sekcija-odsječaka Pri tome u svakoj sekciji mora biti naznačen broj koraka, veličina koraka i trajanje koraka Time se mogu postići složene ovojnica

Na slici 1 se može vidjati zvučne ovojnica istog oblika kao što je npr ima planino ili neki slični instrumenti — brzi početni uspon od nule do pune jačine potom jačine brzo opada na neku srednju vrijednost koja se zadrži neko vrijema a potom jačine polako slabi do nule U stručnoj literaturi se takve ovojnica naziva ADSR ovojnicom (attack decay sustain raleasenagli skok, propadanje održavanje opuštanje)

U praksi to preme slici 1 izgleda Zvuk skače u pet stotinki sekunde na svoju punu jačinu To zahtjeva se 15 nivoa jačine pet koraka minimalnog trajenja te veličinu koraka od +3 Prvi dio instrukcije ENV ovdje ime oblik

ENV 1 5, 2, 3, 1

Sljedećih 20 stotinki sekunde jačina tona pada za jedinice što zahtijeve četiri koraka od 5 štotinki sekunde pa jačina opade za jedinicu po koraku

Naredna sekcija trajanja 50 stotinki sekunde ime samo jedan korak u kojem ne postoji nikakav pad jačine tona Posljedni četvrti korak obuhvaća jedanaset pojedinačnih koraka koji predstavljaju padove jačine zvuka Svaki traje 10 stotinki sekunde Cijela bi se ovojnica dala definirati nstrukcijom

```
ENV 1, 5, 3, 1, 4, −1, 5, 1, 0, 50 11
−1 10
25 READ A, B
35 IF A=0 THEN 75
45 SOUND 1, A B, 0, 1 0, 0,
55 GOTO 25
65 DATA 239, 50, 213 50 190 50
        179 50 159 50, 142 50 127
        50 119 185 0 0
```

Vrijednosti u instrukciji DATA predstavljaju parove nota i dužina trajanja tonova Parametar jačine tona ne traba biti uključen u instrukciju DATA

Važna činjenica je ta da svako trajanje pojedine nota mora biti određeno u instrukciji SOUND To je moguće samo onda ako je trajenje naznačeno u instrukciji SOUND manja od postavljenog u instrukciji ENV ili ako ENV daje konačnu vrijednost jačina tona različitu od nule

## ENT

Instrukcija ENT obavlja svoju funkciju na sličan način kao i ENV Prvi parametar je broj od 1 do 15 koji identificira instrukciju ENT se poziva instrukcijom SOUND, točnije njezinim šestim parametrom Kod instrukcije ENT ponovo se koriste tri broja za definiranja karakteristike svake sekcija razvoja ovojnice a može se koristiti do 5 sekcija Prvi broj postavlja broj koraka (0 do 239), drugi postavlje broj visine tonova u svakom koraku (−128 do +127), a treći upravlja trajanjem svakog koraka (0 do 255) prema jednoj stotinki sekunde

Može se prikazati primjer koji oponaša zvuk policijske strane

```
15 ENT −1, 20, 2, 1, 20, −2, 1
25 SOUND 1 50, 1000, 8 0, 1, 0
```

Prvi broj je negativan, što znači da daje automatsko ponavljanja zvučne ovojnice



Pri korištenju pozitivnog broja ovojnica se izvršava samo jednom a visina tona ostaje na svojoj konačnoj vrijednosti sve dok je određuje instrukcija SOUND Vrijednost u instrukciji ENT daje 20 povišenja visine tona koji buda povećan za 2 pri svakom koraku Svaki korak ime minimalno trajanje od jedne stotinke sekunde Tri naredna parametra, jednostavno okreću taj proces odnosno visinu tona vraćaju na originalnu veličinu Inicijalna veličina visina tona je postavljena veličinom visina tona u instrukciji SOUND U tom slučaju veličina visine tona se kreće prema gore do 40 a potom dolje sa 40 naniže dajući potreban kontinuitet snižavanje i povišavanja visina tona Ako je parametar vremena trajanja korišten u SOUND instrukciji prekratak za tvorbu ovojnica tonove, signal je unatoč tome određan nakon definiranja trajanja u instrukciji SOUND ENV i ENT je moguće zajedno koristiti se instrukcijom SOUND tako da npr prijašnji program može biti nadopunjan linijom 10 i nešto izmenjenom linijom 45

```
10 ENT −1 1, 1 4 2 −1 4 1 1 5
45 SOUND 1 A B 0 1 1 0
```

Instrukcija ENT na jednostavan način varira visinu tona se plus i minus 1 dajući vrlo dobar efekt vibrata

Ovo je bilo osnovno upoznavanja s instrukcijama za korištenja zvuka se računala amstrad/schneider a za potpuno savladavanje tih zvučnih mogućnosti poželjno je koristiti što više literatura i dobrih profesionalnih programa

*Zvonimir Vistrička, dipl. ing*

Put u središte ROM-a

# procedure u akciji

Logo je jezik kojim treba da se služe najmlađi, pa je i definisanje procedura jednostavno. Navodimo najpre službenu reč TO, zatim ime procedure i u sledećim redovima njenu definiciju koju završavamo sa END. Procedura za iscrtavanje kvadrata bi, na primer, mogla da se napiše kao

```
TO drawsquare
PENDOWN
REPEAT 4
[FORWARD 100 LEFT 90]
END
```

Procedura se poziva jednostavnim navođenjem imena: sa drawsquare ćemo navesti računar da iscrta kvadrat počevši od tekuće pozicije „puža", dok ćemo uz tek malo složeniji program dobiti prilično efektnu sliku 1.

slika 1:

```
REPEAT 12
  [ DRAWSQUARE  LEFT 30 ]
```

### Definisanje procedura

Da bismo na što efikasniji način implementirali procedure, podsetićemo se našeg ranijeg napisa „Lične stvari bejzika" u kome smo opisali različite mogućnosti generisanja tabela sa imenima promenljivih. Potrebno je najpre oformiti alfanumerički niz *procname* u koji ćemo smeštati imena potprograma; niz ćemo dimenzionisan tako da „pamti" najviše 30 četvoroslovnih imena. Zašto samo toliko? Verujemo da se program koji razvijamo može veoma zgodno koristiti za mala crtanja, ali da bi za ozbiljnije primene trebalo kreirati editor, potprogram za snimanje na disk i slična modula koja ćemo ostaviti za neku drugu priliku. Ako neko želi duža imena procedura ili promenljivih (imena i u ovoj varijanti programa mogu da budu proizvoljno dugačka, ali se procedura RAČUNARI neće razlikovati od procedure RAČUNICA), jednostavno će promeniti vrednosti konstanti *no_par* i *syml*.

Osim imena procedura treba nekako da pamtimo i tekstove naredbi koje ih realizuju. Za to je rezervisan niz *proclin* u koma su jedna za drugom smeštene sve deklarisane procedure. Kako će računar znati koja procedura počinje od koje linije *proclin*-a i na kojoj se njegovoj liniji završava? Nizovi *start* i *fin* nose odgovarajuće informacije prema šemi sa slike 2.

slika 2:

| procname | start | fin |
| --- | --- | --- |
| DRAWSQUARE | 1 | 2 |
| DRAWHOUSE | 3 | 5 |
| ⋮ | ⋮ | ⋮ |

proclin

| |
| --- |
| "PENDOWN" |
| "REPEAT 4 [FORWARD 100 LEFT 90] !" |
| "DRAWSQUARE FORWARD 100!" |
| "LEFT 30 FORWARD 200!" |
| "LEFT 30 FORWARD 200!" |

Potrebna nam je najzad i informacija o broju trenutno definisanih procedura kako bismo znali kada da prijavimo grešku i gde da upišemo definiciju novog potprograma, uvešćemo, zato, celobrojnu promenljivu *lastpr* kojoj ćemo na početku izvršavanja glavnog programa dodeliti vrednost 0 a zatim je uvećati kada se god definiše nova procedura. Ne bi bilo teško uvesti logo naredbu za brisanje definicije neke procedure, ali nam se čini da bi ona i onako podugačak program načinila još dužim bez nekog stvarnog efekta.

Najjednostavniji način da obezbedimo deklarisanje procedura je dodavanje naredbe *if symbol='TO' then define* u proceduru PLINE, tako će simbol TO na početku svakog reda izazvati poziv procedure *define* koja će primati naredbe sa tastature i smeštati ih u niz *proclin* sve dok ne učita naredbu END kada će adrese početka i kraja definicije upisati u *start* i *fin*. Program *define* je dat na slici 3.

slika 3:

```
[illegible]
```

Pošto smo definisali proceduru, potreban nam je neki način da je pozovemo. Zato ćemo modifikovati proceduru *command* prema slici 4. Pošto nije prepoznao nijednu od standardnih logo naredbi interpretator će umesto da prijavi grešku, pretražiti tabelu imena procedura i pokušati u njoj da pronađe unesenu naredbu. Ako pretraga bude uspešna, procedura *lookup* će dodeliti flag-u *procf* vrednost *true* i tako izazvati poziv procedure *call* i izvršavanje naredbe.

Realizovati potprograme, dakle, nije teško, ali se mora paziti na mnoge finese. Pretpostavimo da je korisnik otkucao REPEAT 6 CRTAJ LEFT 90 gde je CRTAJ ime ranije definisane procedure. Ukoliko procedura CRTAJ ispiše svoju prvu naredbu u bafer *line*, naredba LEFT 90 će jednostavno biti prebrisana i neće moći da se izvrši

slika 4:

```
[illegible]
```

***Verujemo da ste tokom protekla dva meseca otkucali i proučili jednostavni logo interpreter kojim smo ilustrovali naše teorijske napise iz serije „Put u središte ROM-a". Ako ste logo interpreter koristili i za „pravo" crtanje, svakako ste poželeli da ga opremite mnogim „de luks" dodacima. Jedan od takvih dodataka je mogućnost pozivanja procedure i definisanja njihovih argumenata.***

po izlasku iz potprograma Još či veći problem nestati ako jedna procedura poziva drugu druga treću i tako dalje Problem smo rešili definisanjem **lokalnih** promenljivih *line lp count procf* i *proc* u potprogramima *call* i *lookup* Korišćenje lokalnih promenljivih i rekurzija koja su skrivana u pozivima mnogih procedura su osnovni razlog koji nas je naveo da logo interpreter realizujemo na paskalu iako bi bejzik bio u prednosti zbog jednostavnijeg rada sa alfanumericima program pisan na njemu bi bio mnogostruko duži zbog simulacija rekurzijama neophodnog steka

## Procedure sa parametrima

Procedura koje smo do sada pisali se uvek izvršavaju na isti način možemo da kažemo *drawsqure* ali ne i da zadamo dimenzije kvadrata koji treba nacrtati! Zar ne bi bilo lepo definisati proceduru *recktangle* kao na slici 5 a zatim primenom jednostavnih programa dobiti crteže sa slika 1?

namenjen početnicima) kojima prethodi oznaka : U okviru definicije procedure ti se formalni argumenti mogu proizvoljno koristiti sve dok svakom od njih prethodi već pomenuta dvotačka Pri pozivu procedura se iza imena navode stvarni argumenti razdvojani blankovima bez vodećih dvotački Nama naravno nikakvih problema da se iza poziva procedura i lista parametara naveda neka nova naredba koja po potrabi može da bude i poziv novog potprograma

Za uspešnu realizaciju prenošenja parametara u potprograme treba rešiti nekoliko važnih probleme Pre svega, interpreter mora da zna koliko neka procedura ima parametara kako bi po njenom pozivu mogao da pokupi potreban broj simbola iz komandne linije Osim toga, vrednost svakog stvarnog argumenta mora da buda povezana sa imenom odgovarajućeg formalnog argumenta kako bi se u okviru procedura pozivale na ispravnom mestu Rešenja prvog problama su novi nizovi

```
TO RECTANGLE :LENGTH  WIDTH
   PENDOWN
   REPEAT 2 [FORWARD :LENGTH LEFT 90
             FORWARD  WIDTH  LEFT 90]
END
```

slika 5:

slika 6.

```
REPEAT 8 [RECTANGLE 200 100 LEFT 45]
```

```
REPEAT 6[RECTANGLE 200 100 LEFT 30
         RECTANGLE 100 200 LEFT 30]
```

Vidimo da se pri definisanju procedura iza njenog imena navode imena formalnih argumenata (ako ne možete da se setite razlike između stvarnih i formalnih argumenata pogledajte napis Funkcija potprogrami i procedura iz Računara 7' — neka vas ne uvredi činjenica da je tekst

*params paramstart* i *paramname* Da počnemo sa kraja u stringova *paramname* se upisuju imena parametara, *paramstart(i)* ukazuje na ime prvog parametra procedura čiji je redni broj *i* dok *params(i)* čuva ukupan broj njenih parametara Ako vam je ova rečenica izgledala previše komplikovano pogledajte sliku 7

Najpre ćemo napisati proceduru *getparnam* sa slika 8 Ovaj potprogram će naravno pozivati procedrua *define* koje kao što

slika 7




znamo unosi podatka o definisanoj proceduri u odgovarajuća tabela

slika 8:

```
procedure getparnam(var line:str;
                    var lp:integer);
begin
  params[lastpr]:=0;
  getsy(line,lp,symbol);
  if symbol[1]=':'
  then
    begin
      paramstart[lastpr]:=lastpn+1;
      repeat
        params[lastpr]:=
              params[lastpr]+1;
        getsy(line,lp,symbol);
        lastpn:=lastpn+1;
        paramname[lastpn]:=symbol;
        getsy(line,lp,symbol);
      until symbol[1]<>':'
    end
end;
```

Daleko je složenije realizovati poziv procedura Kako smo želeli da jedan potprogram može da poziva drugi i da mu nezavisno prenosi argumente koji mogu da imaju potpuno jednaka imena morali smo da primenimo mehanizam steka Pri svakom pozivu neke procedura u niz *vars* se upisuju imena njenih formalnih argumenata dok se u *vals* upisuju vrednosti stvarnih Svako docnija referenciranja nekog formalnog argumenta će preći preko funkcija *varval* koja će pronaći i vratiti njegovu vrednost Konačni logo interpreter (ili bolje rečeno, logo interpreter koji mi proglašavamo konačnim) dat je na slici 9

Posle prvih nastavaka „Radionica logičkih igara" i „Putovanja u središte ROM-a" autor ovoga teksta je bio u prilici da čuje mnogo korisnih komentara o formi prezentacije softvera koji objavljujemo Pre svega naši programi zaista imaju premalo komentara i nisu dovoljno nazubljeni što se može opravdati samo na jedan način prostor u Računarima je toliko ograničen da na možemo da objavimo duže listinge opterećene ponekom veoma dugom ili veoma uvučenom linijom Druga primedba smo u prilici da uvažimo naš konačni logo inter-

```
Z80-Pascal compiler V 3.00

 - {$C+500}
 - program logo(input,output);
 - {
   C
   C     L O G O    interpretator
   C        grafička naredba
   C     Dejan Ristanović 1986
   C         "Računari  12"
   C
   C }
 - const buff=256; sssl=80; syml=30; psl=9;
 -       proglen=100; nr_var=100; nr_par=30;
 -       a0=48; a9=57;
 - type  str=packed array [1..buff] of char;
 -       word=packed array [1..syml] of char;
         slsl=packed array [1..sssl] of char;
         stsl=packed array [1..psl] of char;
 - var   line,dl:sstsln;
 -       symbol,blword: word;
         i,j,lp:integer;
         x,y,angle,xdir,ydir:real;
 -       gen failed:Boolean;
 -       procsadr: array [1..nr_par] of word;
 -       start:array [1..nr_par] of integer;
 -       fin :array [1..nr_par] of integer;
         program:array [1..proglen] of sln;
 -       lastgn,last:integer;
 -       parnam:array[1..nr_par] of integer;
 -       names:array [1..nr_var] of word;
 -       vals:array [1..nr_var] of integer;
 -       stack,lastst:integer;
 -       parameter:array[1..nr_par]
                   of integer;
 -       pnamename:array[1..nr_par] of word;
 -
 - function upcase(c:char):char;
 - begin
 -    if (ord(c)>96) and (ord(c)<123)
 -    then upcase:=chr(ord(c)-32)
 -    else upcase:=c
 - end;
 -
 - procedure getsy(    line:sln;
 -                 var lp:integer;
 -                 var symbol:word);
 - var j: integer;
 -
 -    function delim(c:char):Boolean;
 -    begin
 -       delim:=(c='[') or (c=' ') or
 -              (c='(') or (c=')') or
 -              (c='=')
 -    end;
 -
 - begin
 -    while line[lp]=' ' do
              lp:=lp+1;
 -    j:=1;
 -    if delim(line[lp])
 -       then begin
 -               symbol[1]:=upcase(line[lp]);
 -               lp:=lp+1
 -            end
         else
            begin
 -             repeat
 -                symbol[j]:=upcase(line[lp]);
 -                lp:=lp+1;
 -                j:=j+1
 -             until delim(line[lp-1]);
 -             lp:=lp-1;
 -             symbol[j-1]:=' '
 -          end
 - end;
 -
 - procedure getline(    prompt:slsl;
 -                   var line:sln);
 - var i,a:integer;
   begin
 -    for i:=1 to pnl do write(prompt[i]);
 -    i:=1;
 -    repeat
 -       repeat
 -          a:=inkey(&1000)
 -       until (a>31) or (a=13);
 -       if (a<>127) and (i<buff)
 -          then
 -             begin
 -                line[i]:=chr(a);
 -                write(line[i]);
 -                i:=i+1;
 -             end
 -          else if a=127 then { DELETE }
 -             begin
 -                if i>1 then
 -                   begin
 -                      write(chr(a));
 -                      i:=i-1
 -                   end
 -             end
 -          else write(chr(7)) {BEEP+greska slova}
 -    until a=13;
 -    line[i-1]:=' ';
 -    writeln
 - end;
 -

 - procedure fail(error:integer);
 - begin
 -    writeln('Error! ',error:3);
 -    failed:=true
 - end;
 -
 - function varval(var line:sln;
 -                 var lp:integer):integer;
 - var nenfnd:Boolean;
 -     sp,i:integer;
 -     ds1,ds2:stsl;
 - begin
 -    getsy(line,lp,symbol);
 -    if stack=0 then
 -                begin fail(15);varval:=0 end
 -    else
 -       begin
 -          sp:=lastst+1;
 -          repeat
 -             sp:=sp-1;
 -             varfnd:=true; i:=1;
 -             while (i<=syml) and nenfnd and
 -                   (symbol[i]<>' ') do
 -             begin
 -                if names[sp,i]<>symbol[i]
 -                then nenfnd:=false;
 -                i:=i+1
 -             end
 -          until nenfnd or (sp=1);
 -          if nenfnd then varval:=vals[sp]
 -          else begin fail(5); varval:=0 end
         end
 - end;
 -
 - function value(var line:sln;
 -                var lp:integer):integer;
 - var s,i,t:integer;
 -     corr :Boolean;
 - begin
 -    getsy(line,lp,symbol);
 -    if symbol[1]=':'
 -    then value:=varval(line,lp)
 -    else begin
 -       i:=1; corr:=true;
 -       while (symbol[i]<>' ') and corr do
 -          begin
 -             t:=ord(symbol[i]);
 -             if (t<a0) or (t>a9)
 -             then corr:=false;
 -             i:=i+1
 -          end;
 -       if corr then s:=val(symbol)
 -       else         s:=0;
 -       if s=0 then fail(2);
 -       value:=s
 -    end
 - end;
 -
 - procedure getparam(var line:sln;
 -                    var lp:integer);
 - begin
 -    parnam[lastgn]:=0;
 -    getsy(line,lp,symbol);
 -    if symbol[1]=':'
 -    then
 -       begin
 -          paramcount[lastgn]:=lastst+1;
 -          repeat
 -             parname[lastgn]:=
 -                  parnam[lastgn]+1;
 -             getsy(line,lp,symbol);
 -             lastgn:=lastgn+1;
 -             pname[lastgn]:=symbol;
 -             getsy(line,lp,symbol)
 -          until symbol[1]<>':'
 -       end
 - end;
 -
 - procedure define;
 - var i:     integer;
 -     dums: sln;
   begin
 -    getsy(line,lp,symbol);
 -    lastgn:=lastgn+1;
 -    procname[lastgn]:=symbol;
 -    getparam(line,lp);
 -    start[lastgn]:=last+1;
 -    getline('?0 line: ',line);
      repeat
 -       last:=last+1;
 -       program[last]:=line;
 -       getline('?0 line: ',line);
         for i:=1 to sssl do
 -             dums[i]:=upcase(line[i])
 -    until dums='END';
 -    fin[lastgn]:=last;
 - end;
 -
 - procedure clear;
 - begin
 -    hdn(0); { brise graficki ekran }
 -    x:=442; y:=578;
 -    plot(4,trunc(x),trunc(y));
 -    { kornjaca na sredini ekrana }
 -    xdir:=0.0; ydir:=1.0; angle:=90;
 -    pen:=true
 - end;
 -
 - procedure proc(    lesslen;
 -                var lp:integer;
 -                var line:sln;
 -                var symbol:word);
 -
 -    procedure command(stand;var line:sln;

                               var lp:integer);
      var end:sln;
 -        i,proc:integer;
          error: Boolean;
 -
 -    procedure rep;
 -    var times,j,long:integer;
 -        procedure doloop;
 -        begin
 -           repeat
 -              lp:=lp+1;
                lp:=inil;
 -              getsy(line,lp,symbol);
 -              command(' ',lp,line,symbol)
 -           until lp>=sssl or failed
 -        end;
 -
          begin
 -           times:=value(line,lp);
 -           if not failed then
 -              begin
 -                 getsy(line,lp,symbol);
                   if symbol[1]<>'['
 -                 then fail(3) else
 -                 begin
 -                    inil:=lp;
 -                    long:=j;
 -                    doloop
 -                 end
 -              end
 -        end;
 
 -    procedure fd_bk(d:integer);
 -    var n:integer;
 -       procedure mak;
 -       begin
            x:=x+d*n*xdir;
 -          y:=y+d*n*ydir;
 -          if pen
 -             then
 -                plot(4,trunc(x),trunc(y))
 -             else
 -                plot(5,trunc(x),trunc(y))
 -       end;
 -
 -    begin
 -       n:=value(line,lp);
 -       if not failed then mak
 -    end;
 -
 -    function rad(x:real):real;
 -    begin
 -       rad:=x*3.1415927/180
 -    end;
 -
      procedure lt_rt(d:integer);
 -    var t:integer;
 -    begin
        t:=value(line,lp);
 -      if not failed
 -         then
 -            begin
 -               angle:=angle+d*t;
                 while angle>360 do
                       angle:=angle-360;
 -               while angle<0 do
 -                     angle:=angle+360;
 -               xdir:=cos(rad(angle));
                 ydir:=sin(rad(angle))
 -            end
 -    end;
 -
 -    procedure look(var procf:Boolean);
 -    var ds1,ds2: stsl;
 -        i       : integer;
 -    begin
        if lastgn=0 then procf:=false
        else
 -       begin
 -          procs:=0;
 -          repeat
 -             procs:=procs+1;
               for i:=1 to sssl do
 -             begin
 -                ds1[i]:=symbol[i];
 -                ds2[i]:=procname[procs,i]
 -             end;
               procf:=ds1=ds2
 -          until procf or (procs=lastgn)
 -       end
 -    end;
 -
 -    procedure getpar(    line:sln;
 -                     var lp:integer);
 -    var pn,nexlen,v:integer;
 -    begin
        if parnam[procs]<>0
        then begin
 -          pn:=parcount[procs];
            nexlen:=pn+parnam[procs];
 -          repeat
 -             v:=value(line,lp);
               stack:=stack+1;
 -             names[stack]:=
 -                   pnamename[pn];
 -             vals[stack]:=v;
 -             pn:=pn+1
 -          until (pn=nexlen) or failed
 -       end
 -    end;
 -
 -    procedure call(procs:integer);
 -    var line: sln;
 -        lp,dens: integer;
```

```
        until false
      end
0 Compilation error s
Code size = 4305 bytes
```

preter intenzivno koristi pakovane matrice za smeštanje stringova (raspitali smo se okolo i saznali da paskal kompajleri svih popularnih personalnih modela podržavaju ovu strukturu) koristićemo FORWARD reference funkcija smanjili smo upotrebu globalnih simbola i prenosili argumente gde god je to ispistivo sve smo veličine koje ima smisla menjati deklarisali kao konstante da bi ih čitaoci lakše menjali (nije prijatno loviti' dužinu nekog stringa ili maksimalnu dimenziju matrice po čitavom programu) Promenili smo najzad funkciju *value* tako da pre poziva funkcije *ival* ispita string koji treba pretvarati u broj i ako u njemu nađe slova prijaviti grešku Na taj se način izbegava da program prekine izvršavanje i prijavi grašku Bad value baš kad ste otkucavši LEFT FORWARD 100 nameravali da iscrtate poslednju liniju na nekoj komplikovanoj slici

Završavajući prezentaciju logo interpretera reći ćemo nekoliko reči o načinima na koje bi on mogao da se učini još boljim Trebalo bi pre svega omogućiti unošenje logo programa u memoriju i njihovo snimanje učitavanje i editovanje Interpreter bi trebalo dopuniti modulima za izračunavanje vrednosti aritmetičkih izraza (time ćemo se baviti već u sledećem nastavku ove Škole) i rad sa stringovima, a mogle bi se napraviti i nove kontrolne strukture Knjiga Advanced Programminig Tiechniques Džima Mekgregora i Alena Vata (Jima Mc Gregora i Alena Watts) koju upravo naputštamo govori o dodavanju naredbe STOP koja izaziva izlazak iz procedure i omogućava izuzetno efektne programa poput onoga sa slike 10

```
slika 10:

TO spiral :len :wid
   IF (:wid<4) THEN [STOP]
   rectangle :len :wid
   LEFT 10
   spiral (:len-4) (:wid-4)
END
```

## *Upotreba jezičkih procesora*

Logo interpretator nije samo ilustracija raznih oblasti sistemskog programiranja koje smo pominjali u ovoj Školi, on je i dobar primer jezičkog procesora Zamislite da je neko naručio složeni programski sistem za crtanje ili CAD (Computer Aided Design) Ovakvi se paketi obično prave tako da specijalni tasteri vašeg računara dobiju razne funkcije i da se po izboru neke od njih preko tastature unose potrebni parametri iako potencijalno veoma moćni ovakvi su programski paketi teški za upotrebu Korisnik pre svega mora da zapamti funkciju svakog tastera zatim da po pritisku na njega odgovara na niz pitanja i da najzad posle svake greške preduzima komplikovane manevre (koje takođe treba znati napamet) da bi izbrisao neki element ne oštetivši ostatak Drugi pristup problemu je nešto teži za realizaciju ali omogućava daleko komforniju upotrebu treba kreirati **programski jezik za crtanje** koji se brzo i lako uči i još lakše upotrebljava Budućem će korisniku biti daleko lakše da zapamti desetak reči koje imaju smisla nego da memoriše funkcije mnoštva specijalnih tastera Osim toga korisnik će biti opušten u radu ako zna da svaku grešku neće platiti ponovnim crtanjem slike posle izmene neka naredba logo programa računar će samostalno i veoma brzo ponoviti čitav crtež

Pisanje jezičkog procesora je još prosperitetnija aktivnost kada se piše neki program paket koji obavlja složene matematičke ili inženjerske proračune napisaćemo standardni interpretator koji podržava naredba za izračunavanje izraza unošenje i dodeljivanje vrednosti kao i osnovne kontrolne struktura (IF THEN ELSE WHILE END WHILE GOTO ), a zatim uvesti naredbe jasno izvršavati relativno dugo ali to ne treba da nas zabrinjava, korisnik će i dalje dobiti odgovor nekoliko miliona puta brže nego da je radio na ruke ili koristio neki kalkulator Sa druge strane korisnik našeg programskog paketa neće morati da opterećuje svoju memoriju stvarima koje mu nisu potrebne pamtiće samo naredba koje koristi u svakodnevnom radu Sve ostale informacije o korišćenju programskog paketa se mogu naći u okviru uputstva za upotrebu i to u vrlo kratkom vremenu obavezno priložimo abecedni spisak na kome je svakoj naredbi posvećena po jedna (ili po nekoliko) strana sa detaljnim opisom

Recimo na kraju, da je pisanje specijalizovanih programskih jezika povoljna varijanta čak i za autore softvera jeste da je teže napisati interpreter nego program koji radi sa menijima ali se čitav posao radi jedanput u životu kada smo razvili opšti deo jezika posao smo naravno obavili tako da jezik bude okrenut proširenjima i dobro dokumentovan) za sva ostale programske pakete koje proizvodimo ćemo samo koncipirati specijalizovane naredba i dodavati module koji ih izvršavaju

***Dejan Ristanović***

## NASTAVAK SA 43 STRANE

## *sam svoj programer*

Programi koji koriste EXOPEN treba ovaj broj da sačuvaju i pri kasnijem obraćanju periferiji da ga koriste kao logički broj veze što je već standardan način Određivanje logičkog broja od strane programa umesto da to čini korisnik ima i jednu prednost ne može se slučajno upotrebiti već upotrebljen broj čime otpada greška broj 2 EXOPEN u tablici logičkih adresa već otvoranih veza pronalazi najveću i zatim kao novi logički broj uzima jedan veći Tako će prva otvorena veza uvek imati broj 1 sledeća broj 2 itd Ako se zatim zatvori samo broj 1 a otvori nova veza njen broj će biti 3 iako 1 više ne postoji Ostatak rutine teče uobičajeno traži se bajt nule na kraju naziva da bi se odredila njegova dužina izdvajaju iz akumulatora prva i sekundarna adresa i pozivaju standardne KERNAL rutine ali i funkcije broj 40 (IOERROR) ako je veza otvorena prema disku da bi se proverilo njegovo stanje Obratite pažnju kako je obezbeđeno stanje CARRY zastavice na izlazu

Evo i nekoliko predloga za rutine koje treba razviti Pošto nema oslanjanja na bejzik ROM nedostajaće rutine za ispis bilo koje poruke na primer SPRINT (String Print) čiji bi ulazni parametri bili početna adresa poruke a kraj može biti obeležen bajtom 0 isto važi i za SINPUT (String Input), koji može imati kao ulazne parametre takođe početnu adresu od koje treba upisati liniju, dok se krajem linije smatra terminator 13 koji rutina treba da zameni sa 0 kako bi se održala kompatibilnost sa SPRINT Vreme je da pokrenemo i pitanje grafike visoke rezolucije sa recimo rutinama MODE (izbor jednog od tri video moda) PALETTE (postavljenje boja koje će biti korišćene) i PLOT (setovanje/resetovanje jedne tačke na ekranu visoke rezolucije) Ovo su relativno jednostavni zadaci ali baš zato zahtevaju da budu i maksimalno efikasno rešeni Kada to bude urađeno, možemo da vam uputimo i prvi izazov napravite rutine SPRINT i SINPUT koje će razlikovati u kom modu je video procesor (tekst ili visoka rezolucija) ali će i dalje obavljati istu zadatak bez promene moda Ako sve to bude realizovano konačna proširenja bejzika će biti vrlo lako realizovati a naš operativni sistem će moći da ponese i bar deo prvog slova atribut GRAFIČKI

***Zoran Životić***

# Pokice koje život znače

Da bi vam malo pomogli u igranju bezbrojnih igara koje upravo zagušuju tržište evo vam prava gomila pokova koji će vam pomoći da lakše stignete do kraja. Najveći deo pokova daje bezbroj života dok neki rešavaju neke konkretne prepreke u igri. Koliko igara svakog dana stiže u Jugoslaviju teško da bi imali vremena da ih sve igrate do kraja da nema raznih pokova, zar ne? Izvolite

| | |
|---|---|
| Arcadia | POKE 23778 0 III POKE 25778,0 |
| Cookie | POKE 35730 52 |
| Ground Attack | POKE 29063 0 |
| Hunchback | POKE 28888 0 |
| Jet-Pack | POKE 25020 0 |
| Kokotoni Wilf | Prvo treba uneti sledeći programčić i izvršiti ga pa zatim uneti igru<br>10 CLEAR 24100 LOAD CODE<br>20 RANDOMISE USR 65100 LOAD CODE<br>30 POKE 42214 X PRINT USR 41200<br>gde je X=broj života |
| Kosmic Kanga | POKE 36212 0 |
| Lunar Jetman | POKE 36965,0 (bezbroj života)<br>POKE 37599 201 (brisanje svih neprijatelja) |
| Mr Wimpy | POKE 33053 0 |
| Mutant Monty | POKE 54933 0 (to vam daje 255 života) |
| Pibailed | POKE 46441 0 |
| PSSST | POKE 24964 0 |
| TranzAm | POKE 25446 0 |
| Freez Beez | POKE 34610 0 |
| Pyramid | POKE 44685 0 |
| Sooty | Uneti i startovati sledeći program pre igre<br>10 BORDER O PAPER O INK O<br>20 CLEAR 26870<br>30 LOAD SCREEN$<br>40 LOAD CODE 26880<br>50 RANDOMISE USR 26880<br>60 POKE 58294 0<br>70 RANDOMISE USR 52500 |
| Spectral Invaders | POKE 25062 254 |
| Cauldron | Uneti i startovati neposredno pre igre<br>10 CLEAR 24599 FOR f=23296 TO 23308 READ a POKEf a NEXT f<br>20 LET L=USR 23296<br>POKE 40060 0 LET L=USR 24600<br>30 DATA 221 33 24 96 17 232 159 62 255 55 205 86 5 201 |
| River Rescue | POKE 33420 0 (prvi igrač)<br>POKE 33452 0 (drugi igrač) |
| Zaxxon | POKE 46825 255 |
| Cavern fighter | POKE 31683 0 POKE 31684 0 |
| BLack Hawk | POKE 34885 183 |
| A View to a Kill | Dodati liniji 30 ispred naredbe Rand USR POKE 28032 255 POKE 28087 8 POKE 29243 6 POKE 30543 8 |
| Project Future | POKE 27682 2 |
| Road Racer | POKE 27150 0 |
| Ah Diddums | POKE 24786 0 |
| Orion | POKE 37318 201 |
| Zip Zap | Poke 54085 0 |
| Jet Set Willy 2 | Uneti i startovati pre igre<br>10 Clear 64998<br>20 Let obj=150<br>30 Let room=32<br>40 For n=65000 To 65047 READ a.POKE n a NEXT n<br>50 PAPER O INK O BORDER O CLS<br>60 RANDOMISE USR 65000<br>70 DATA 221 33 0 64 17 56 185 62 255 55 205, |

## KOMODOR

| | | POKE |
|---|---|---|
| Annihilator | | 6295,11 |
| Attack of the mutant camels | | 11639,255 |
| Apshay, Gateway of | | 2264,99 |
| Blagger | | 3560,0 |
| Blagger | | 53264,126 |
| Blagger | | 3574,44 |
| Burnin rubber | | 18432,173 |
| Bruce Lee | 1. | 5677,128 |
| Bruce Lee | 2 | 5686,128 |
| Bruce Lee | | 7828,165 |
| Bruce Lee | | 5472,99 |
| Bruce Lee | | 5672,128 |
| Bagitman | | 19013,189 |
| Bagitman | | 53264,126 |
| Bagitman | | 22236,255 |
| Bagitman | | 22235,5 |
| Bagitman | | 22236,39 |
| Back Rogers | | 8852,36 |
| Back Rogers | | 2490,9 |
| Black hawk | | 8290,255 |
| Black hawk | | 8289,99 |
| Black hawk | | 2771,238 |
| Battle trough time | | 22045,255 |
| Battle trough time | | 16877,234 |
| Bungling bay | | 47465,176 |
| Bat attack | | 11861,234 |
| Battle zone | | 8989,188 |
| Boulder dash | | 16494,165 |
| Crazy kong | | 38624,173 |
| Cross fire | | 27625,173 |
| Cross fire | | 5353,44 |
| Choplifter | | 8811,173 |
| China miner | | 33381,245 |
| China miner | | 33457,255 |
| China miner | | 34623,44 |
| China miner | 1. | 34623,234 |
| China miner | 2 | 34624,234 |
| China miner | 3. | 34625,234 |
| Crissis mountine | 1. | 2665,238 |
| Crissis mountine | 2. | 3144,238 |
| Crissis mountine | | 3566,188 |
| Clowns | | 3566,255 |
| Cavelon | | 23789,255 |
| Cavelon | | 33789,95 |
| Cavelon | | 15458,255 |
| Chiller | | 22581,189 |
| Cuthbert in jungle | | 2659,5 |
| Dinky doo | | 11987,99 |
| Dinky doo | | 11989,18 |
| Dinky doo | | 12296,165 |
| Donkey kong | | 12118,234 |
| Dimension X | | 8645,129 |
| Dig dug | | 18473,255 |
| Dare devil Dennis | | 29173,255 |
| Dare devil Dennis (x-brzina) | | 17958,x |
| Dragon hawl | | 3477,255 |
| Defender | | 3885,5 |
| Defender | | 2814,255 |
| Evolutin | | 4947,25 |
| Eagle empire | | 22144,8 |
| Encounter | | 38430,8 |
| Frogger | | 22341,173 |
| Fire ant | | 17568,18 |
| Fire ant | | 17567,7 |
| Fort apocalypse | 1 | 34364,234 |
| Fort apocalypse | 2 | 36365,234 |
| Fort apocalypse | 3 | [illegible] |
| Fort apocalypse | 4 | [illegible] |
| Fort apocalypse | 1. | [illegible] |
| Fort apocalypse | 2. | 14760,8 |
| Fort apocalypse | 3 | 36366,0 |
| Fort apocalypse | | 36337,99 |
| Fort apocalypse | | 36339,153 |
| Falcon patrol | 1 | 16764,36 |
| Falcon patrol | 2 | 16785,2 |
| Falcon patrol | | 16764,234 |
| Falcon patrol (sys 16640) | | 16765,234 |
| Fantastic Fredy | | 31887,255 |
| Fantastic Fredy | | 34535,24 |
| Fantastic Fredy | | 2465,165 |
| Frogger Sega | | 22341,173 |
| Flak | | 4798,36 |
| Galaga | | 17388,173 |
| Galaxians | 1. | 7865,238 |
| Galaxions | 2. | 17288,165 |
| Galaxy | 1 | 3369,238 |
| Galaxy | 2 | 3378,238 |
| Gangster | | 5989,58 |
| Gangster | | 3574,44 |
| Ghouls | | 8367,255 |
| Ghouls | | 8367,99 |
| Gyross | | 3999,255 |
| Gyross | | 3999,288 |
| Gyross | | 4491,59 |
| Hard hat mack | | 16877,188 |
| Hard hat mack | | 8472,188 |
| Hard hat mack | | 16877,234 |
| Hunchback | | 9591,234 |
| Hunchback | 1 | 9521,234 |
| Hunchback | 2. | 9522,234 |
| Hunchback | 3 | 9523,234 |
| Herby | | 7191,255 |
| Hero | | 19131,8 |
| Hero | | 13865,8 |
| Hero | | 19131,8 |
| High noon | | 18833,255 |
| House of usher | | 6721,238 |
| House of usher | | 7870,68 |
| Jumping jack | | 27984,173 |
| Jumpman junior | | 9458,44 |
| Jumpman junior | | 9458,173 |
| Jet Set Willy | | 11345,33 |
| Jungle hunt | 1 | 2242,234 |
| Jungle hunt | 2 | 2243,234 |
| Jungle hunt | | 2242,165 |
| Kickman | | 7424,238 |
| Kaktus | | 4565,255 |
| Kid grid | | 18828,234 |
| Laser strike | | 16475,173 |
| Lazy Jones | | 2971,9 |
| Lodarunner | | 7892,255 |
| Lady Tut | | 2392,58 |
| Lancer lords | | 16424,68 |
| Miner 2049 er | | 9458,173 |
| Miner 2049 er | | 2652,165 |
| Miner 2049 er | | 2652,188 |
| Miner 2049 er | | 2471,255 |
| Miner 2049 er | | 11518,255 |
| Motor mania | | 8646,255 |
| Mrs. Pacman | | 8898,173 |
| Manic miner (sys16384) | | 16571,173 |
| Manic miner (sys16384) | | 15456,75 |
| Manic miner (sys16384) | | 15424,75 |
| Moon buggy | | 25151,173 |
| Moon buggy | | 38624,173 |
| Moon buggy | | 24151,173 |
| Matrix | 1 | 7629,238 |
| Matrix | 2. | 7983,238 |
| Mr Robot & H.F. | | 11518,255 |
| Maggot mania | | 2532,4 |
| Monty mole | | 7863,8 |
| Moon patrol (Atari) | | 2872,255 |
| Moon patrol | | 2872,288 |
| Neptune daughters | | 7870,68 |
| Omega race | | 6388,238 |
| Pooyan | | 28634,173 |
| Pogo Joe | | 23558,169 |
| Pech (Pengo) | | 28295,44 |
| Punchy | | 15458,58 |
| Pedestrian | | 2288,255 |
| Pacacuda | | 7815,234 |
| Pitfall | | 5393,255 |

| | | |
|---|---|---|
| Pitfall | | 2668,250 |
| Popeye | 1. | 2485,255 |
| Popeye | 2. | 2486,255 |
| Protector 2 | | 16425,4 |
| R - bert | | 4446,17 |
| Q bert | | 4446,238 |
| Quest for tires | | 14864,8 |
| Quest for tires | | 7341,99 |
| Quest for tires | | 11485,12 |
| Radar rat race | | 7194,25 |
| Revenge camels | 1. | 2599,250 |
| Revenfe camels | 2. | 2746,238 |
| Revenge camels | 3. | 39931,238 |
| Revenge camels | | 35518,250 |
| R - nest | | 4444,17 |
| R - nest | | 4446,239 |
| R - nest | | 2759,100 |
| Round about | | 12843,234 |
| Robin rescue | | 6144,234 |
| Ring of power | | 43,28 |
| Ring of power | | 44,24 |
| Snookie | | 33242,255 |
| Snookie | | 33242,200 |
| Snookie | | 38116,177 |
| Sea fox | | 7337,17 |
| Sammy lighfoot | | 3678,189 |
| Shamus | 1. | 27185,169 |
| Shamus | 2 | 27185,165 |
| Shamus | 3. | 27185,141 |
| Shamus | 4. | 27185,54 |
| Shamus | 5. | 27185,179 |
| Shamus | 1. | 18406,169 |
| Shamus | 2 | 23558,169 |
| Spelunker | 1. | 19125,234 |
| Spelunker | 2. | 19126,234 |
| Spelunker | | 18487,44 |
| Shamus case 2 | | 15475,238 |
| Shamus case 2 | | 3868,126 |
| Shamus case 2 | | 15476,176 |
| Space taxi | | 16911,200 |
| Sheep in space | | 35839,44 |
| Sqish em | | 2562,100 |
| Skramble (x=brzina) | | 4691,1 |
| Skramble | | 11291,175 |
| Survivor | | 19563,255 |
| Survivor | | 19523,200 |
| Tales of Arabian nights | 1 | 2531,173 |
| Tales Of Arabian nights | 2 | 2632,141 |
| Tapper | | 15899,165 |
| Time runner | | 8343,9 |
| Trollie wallie | | 7711,177 |
| Wheellin wallie | | 27427,17 |
| Wheellin wallie | | 27916,177 |
| Whistler brothers | | 26100,0 |
| Xerone | | 2273,230 |
| Zeppelin | | 18546,44 |
| Zeppelin | | 18881,100 |
| Zeppelin | | 14337,100 |

309 komada pokica na broju

GHOSTBUSTERS:

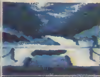
| ime: | novac: | konto: |
|---|---|---|
| peter | 29.900 | 3166438R |
| twerk,the | 40.000 | 86660401 |
| butterfly | 60.000 | 84664701 |
| nnn | 77.000 | 20406201 |
| s,m | 100.000 | 1984 |
| 12345 | 415 500 | 25393120 |
| herbie | 500 500 | 85250624 |
| butterfly | 512 000 | 38542224 |

Pripremio: Vlada Kratanovic

```
06 5 243 40 240
80 DATA 33 0 254 17
107 100 1 50 0 237 176
195 0 95
80 DATA 62 255 50 07
117
100 DATA 62, obj, 50 126
135
110 DATA 62 room 50 75
117
120 DATA 105 0 112
```

57/pokice koje život znače

# Komodor 64



## Winter games

*Zimske Olimpijske igre*

*Evo igre koja će u predelima izvornog novog primitivizma biti izuzetno dobro prihvaćena. Samo imaju li novi primitivci računare? Sve što vam treba je tu: sneg, planine, tereni, pa čak i publika. Hladnoća ne nedostaje. Ali zato nedostaju čevabdžinice.*

*Možete da birate između osamdeset zemalja, a disciplina je samo sedam. Očigledno je da su ljudi iz Dekatlona planirali da im se igra prodaje (pardon, piratizuje) u dosta zemalja, pa su obezbedili boje za domaće. Za nas je ovo verovatno jedini način da osim one Frankove osvojimo još koju medalju. Džojstik se dosta upotrebljava u svakoj igri, tako da je potreban fin osećaj za odskoke, skretanja i slične finese jer nesreće nisu retke. To se najbolje vidi na skokovima gde se lako strada ako odskočite prerano. Meni lično je najzanimljivije klizanje jer neodoljivo podseća na neku bonlačku veštinu. Ono što njemu fali je nazalni glas Milke Babović, ali ništa nije savršeno, zar ne?*

## Racing Destruction set

*Staza uništenja*

*Evo još jedne trkačke simulacije. U ovom slučaju ime vara — ne radi se o trkama na našim putevima, već o pravoj trkačkoj stazi. Igra ne počinje vožnjom. Prvo treba konstruisati stazu, postavljajući ukrštanja, mostove, nadvožnjake i ostale delove staze. Sledeći korak je podešavanje automobila koji želite da vozite. Možete praktično da konstruišete bilo kakvo vozilo koje zamislite. Samo treba odabrati dodatke ili izmene i vozilo je tu. Oni se izrazito mazohističkim svojstvima će sigurno biti obradovani kad saznaju da što su veća i jača kola to se ona lepše skrckaju kad izletite sa staze.*

*Pogled na igru je što je neuobičajeno iz ptičje perspektive. Ekran je podeljen u dva dela, tako da mogu učestvovati i dva igrača. Onaj deo trke koji je najzanimljiviji vozačkom amateru (to sam ja) su skokovi na onim mestima gde ste postavili odskočne rampe. Sve je to lepo i realistično, ali ako ne pazite može postati i suviše realistično — automobilčići izgledaju tako realistično kad su zgužvani u lepte kraj puta.*

# amstrad



## Strip poker

Ovu igru ne treba posebno predstavljati, jer se igrala još kada se o računarima moglo samo maštati. Nemojte misliti da vam je protivnik neki računarski skrep u obliku nekog maštovitog sprajta. Protivnik vam je tako dobro nacrtan da ne treba da se iznenadite ako najveći (muški) deo vašeg komšiluka iznenada dobije želju da isproba vaš „amstrad" i naravno ovu igru. Čuvajte se, vaš protivnik igra suviše dobro i skoro ga je nemoguće pobediti. Zato vi lepo verujte i ne skidajte se dok vaše protivnica nalvne veruje u pravila i posle svake izgubljene partije skida po jedan deo odeće. Ova je igra za koju je najčešće traženo kako stići direktno do kraja bez borbe. Pogodite zašto. Ako vas još interesuje, pišite nam. Do tada puno sreće u pokeru i pazite da ne — nasedete!

## Super Pipeline II

*Super Cevovod II*

*Da igre nemaju granica već znamo. Ovde je to pokazano time što ste nadzornik vodovoda (!?) čiji je zadatak da nadgleda instalacije cevovoda i da napuni par buradi dnevno. Pošto ste nadzornik vi u stvari ne radite (uobičajeno, zar ne?), već imate dva radnika koji popravljaju oštećenja umesto vas. Naravno, tu je gomila gamadi i nevolja koji pokušavaju da vas sreate. Za odbranu imate pištolj (pomalo neuobičajeno za vodovodnog nadzornika) sa kim morate biti vešti kao Klint Istvud ako želite da preživite. Ako baš zagusti, možete da žrtvujete jednog radnika da biste se zaštitili (nije humano, ali je jako korisno). Poseban plus u igri su mali crtači i odlična muzika posle svakog savladanog nivoa. Igrajte i uživajte, čak je i u cevovodu rad ponekad zanimljiv.*

# Hajde da se igramo

# elitna groznica

Iako su verzije „Elite" za BBC, „komodore 64" i „spektrum" dela raznih softverskih firmi, sami programi su veoma slični, što znači da će vlasnici različitih računara manje-više ravnopravno da se koriste savetima koje dajemo, pa čak i da upotrebe program za „varanje" koji pripremamo za sledeće „Računare". Verujući da su uputstva za upotrebu sve tri verzije programa identična, dajemo ih na slici 1.

slika 1

```
E L I T E   Komande

U stanici:

f0:  uzletanje.
f1:  kupovina robe.
f2:  prodaja robe.
f3:  kupovina opreme
f4:  karta galaksije
f5:  karta okoline
f6:  podaci o planeti na koju ste
     pozicionirali kursor
f7:  stan i raspoloživa količina robe
f8:  trenutno stanje broda
     (oprema, rejting, status)
f9:  inventar robe koju brod nosi
CTRL H  pozicioniranje na svet do koga se
     putuje skokom kroz hipersvemir.
O    pozicioniranje kursora u centar mape.
D    rastojanje do sveta na koji ste
     pozicionirali kursor
F    traženje sveta po imenu (samo verzija
@    snimanje rejtinga na traku/disk
COPY  zamrzavanje igre
DELETE  odmrzavanje igre

Dok je igra zamrznuta:

ESCAPE  prekid igre
J :  prebacivanje keyboard/joystick
K :  [illegible] kanala na joysticku
S :  uključivanje zvuka
Q :  isključivanje zvuka

Dok je brod u svemiru

A    laser
T    nišanjenje projektila
M    ispaljivanje projektila
     (samo ako je prethodno nacrtan [illegible]
     ako njegov kvadratić u levom donjem
     uglu ekrana svetli crveno)
U    deaktiviranje [illegible] projektila
E    aktiviranje svih vaših i protivničkih
     projektila u okolini; samo uz ECM
ESCAPE  iskakanje kapsulom za spasavanje
TAB  aktiviranje energetske bombe
C    aktiviranje kompjutera za sletanje
P    deaktiviranje kompjutera za sletanje
S    gore
X    dole
<    rotacija levo
>    rotacija desno
SPACE  gas
?    kočenje
J    skok [illegible]
     samo ako je "Condition Green")
H    skok kroz hipersvemir
CTRL H  skok u drugu galaksiju
f0   pogled napred
f1   pogled nazad
f2   pogled levo
f3   pogled desno
COPY  zamrzava igru
DELETE  odmrzava igru

Ukratko o kontrolnom panelu

Levo gore     stanje prednjih štitnika
Ispod         stanje zadnjih štitnika
Ispod         količina goriva
Ispod         temperatura u kabini
Ispod         temperatura lasera
Ispod         rastojanje do planete
Levo dole     broj raketa koje posedujete
Desno gore    brzina kretanja
Ispod:        [illegible]
Ispod         energija (četiri reda)
Sredina       [illegible]
              [illegible]
              [illegible]
              [illegible]
              [illegible]
              [illegible]
              [illegible]
              [illegible]
              [illegible]
              [illegible]
              [illegible]
              stanici
```

## *Bez alata nema ni zanata*

Na svemirskim stanicama se mogu nabaviti razni uređaji koji povećavaju vatrenu i odbrambenu snagu vaše Kobre 3. Mnogi vlasnici piratskih kopija „Elite", na žalost, ne kupuju ove dodatke jer bez uputstva ne umeju da ih iskoriste. Velika greška: vaš rejting nikada neće bitnije napredovati ako ne opremite vaš arsenal svim postojećim naoružanjem!

U startu ste opremljeni samo slabim 'pulse' (pulsirajući) laserom koji je smešten sa prednje strane vašeg broda. Ukoliko, pritiskajući neki od funkcijskih tastera, pogledate na drugu stranu, videćete da se u sredini ekrana ne pojavljuje nišan: što znači da ne posedujete lasere kojima biste mogli da gađate brodove koji vam prilaze sa strane ili leđa. Takvi se laseri mogu relativno jeftino dobiti. S druge strane, daleko je bolje investirati u snažnije beam (zračni) lasere ili bar u jedan takav uređaj koji ćete staviti sa prednje strane broda — neprijatelji koje ćete pobiti i teret koji ćete zaplenititi će veoma brzo opravdati takvu investiciju.

'Beam' laseri nisu međutim, ni u kom slučaju konačni: za 5000 kredita se može kupiti 'military' laser koji je višestruko snažniji od običnog. Jedan pogodak vojnim laserom će uništiti gotovo svakog neprijatelja! Bez ovakvog se lasera ne možete boriti protiv targoida niti uspešno okončati drugu misiju (kasetna verzija „Elite" za BBC ne podržava misije niti vojne lasere). Čim skupite potreban broj kredita, kupite prednji i zadnji vojni laser. Laseri sa leve i desne strane nisu naročito potrebni, jer je njima skoro nemoguće rukovati.

Sledeća važna investicija je mnogo jeftini 'Energy Unit' (Energetska jedinica). Kada, naime, vaši prednji štitnici budu istrošeni u toku borbe, počeće intenzivno pražnjenje energetskih rezervi broda. Kada se energija iz četiri rezervoara potroši, brod eksplodira, što predstavlja kraj igre. Srećna je okolnost što se energetski rezervoar puni iz spoljnog prostora, što višestruko ubrzava energetska jedinica — bez nje nikako nemate smisla da putujete u svetove u kojima vlada anarhija, jer će piratski talasi navaljivati na vas pre nego što vam se energija dopuni. U sledećim ćemo „Računarima" objaviti program pomoću koga ćete sebi dodeljivati energetsku jedinicu strahovite snage i tako postajati praktično neuništivi.

## *Skupljači goriva i energetske bombe*

'Fuel scoops' (Skupljač goriva) su vredan dodatak koji ćete vrlo retko koristiti za ono za šta je po imenu namenjen. Vlasnik „skupljača goriva" može naime da priđe bilo kom suncu (treba prići toliko da temperatura u kabini dostigne maksimum ali ne više od toga), postaviti se tako da vidi samo vrh zvezde i krenuti napred maksimalnom brzinom. Rezervoari će se ubrzo napuniti i bićete u prilici da kroz hipersvemir otputujete do susedne planete; odredište skoka treba unapred izabrati jer u uslovima visoke temperature nećete moći da mislite na to.

Verujemo da ćete se veoma retko upuštati u sakupljanje goriva: ova je sirovina relativno jeftina i nema je smisla skupljati uz ogroman rizik od pregrevanja. Ovaj ćete metod, dakle, koristiti samo ako ostanete bez goriva u blizini stanice koja šalje policijske brodove za vama i onemogućava vam sletanje. Fuel scoops vam međutim omogućavaju jednu mnogo korisniju stvar: skupljanje tereta koji je ispustio vasionski brod koji ste upravo uništili: svaki kontejner koji „upecate" može da bude bezvredan (ako sadrži hranu ili tekstil) ili vredan čitavo bogatstvo. Uz malo hrabrosti možete da pokupite čak i kapsulu za spasavanje u kojoj se krije pilot neprijateljskog broda: on postaje vaš rob koga možete da prodate na prvoj susednoj planeti.

'ECM system' nije mnogo koristan, ali nije ni mnogo skup. On vam omogućava da uništite sve protivničke (na žalost, i sopstvene) projektile u vidokrugu i tako se iskoćom parirate piratima koji nose dirigovane rakete. S druge strane, dirigovana raketa se, uz malo truda, može zaobići ili uništiti tako da ćete taster E pritiskati samo u trenucima panike kada vas nekoliko neprijatelja gađa projektilima. Direktan pogodak projektila vas neće trenutno uništiti ali će teško oštetiti vaše štitnike ili isprazniti bar tri od vaša četiri energetska rezervoara.

'Energy Bomb' (Energetska bomba) je skup i malo isplativ dodatak koji ćete kupovati tek kada vaše finansije ozbiljno uznapreduju. Svaka energetska bomba može da eksplodira samo jednom i tom prilikom uništi sve neprijateljske brodove, projektile, teret i ostale objekte u vidokrugu.

'Escape Pod' (Kapsula za spasavanje) se koristi samo u specijalnim situacijama. Za trenutak će vas vratiti na stanicu gde ćete od osiguravajućeg društva dobiti nov brod opremljen poput onoga koji ste izgubili; nedostajaće vam jedino „potrošena" kapsula za spasavanje i teret koji ste prenosili. Daleko je pametnije snimiti vaš status na disketu (ili traku) u svakoj stanici i u slučaju pogibije jednostavno ponovo učitati ovaj snimak.

***„Elite", igra koju smo predstavili još u „Računarima 4, nije više privilegija vlasnika BBC-ja i „elektrona" verzije „Elite" za „komodor 64" i „spektrum" su se pojavile sredinom 1985, da bi krajem iste godine, uz oglasne fanfare, pristigle i na jugoslovensko piratsko tržište. Zato smo odlučili da ponovo posvetimo pažnju ovoj fascinantnoj igri i da, kroz napise u ovim i sledećim „Računarima", opišemo kako se stiže do njenog kraja. Pri tom ćemo, naravno, obratiti posebnu pažnju na raznorazna „varanja" koja će vam omogućiti da za veoma kratko vreme i praktično bez napora dostignete magični status „Elite".***

## Kompjuter za sletanje

Docking Computer (Kompjuter za sletanje) je san svih početnika u Eliti da biste trgovali odnosno nećevijali gorivo i opremu morate da slećete na svemirske stanice a pri tim sletanjima redovno ginete! Kupovina kompjutera za sletanje će automatizovati ovaj postupak i smanjiti rizik ali ga neće potpuno ukinuti čak i kompjuter za sletanje vas ponekad može razlupati o stanicu pogotovu ako ga startujete previše kasno Nevolje sa kompjuterom za sletanje je što je veoma skup tako da ćete kada steknete novac za njegovu kupovinu, verovatno već savršeno dobro sletati i bez njega ipak ga kupite videćete pre svega kako kompjuter upravlja brodom a i čuti lepu interpretaciju kompozicije „Na lepom plavom Dunavu" samo verzije za „komodor 64" Dužni smo vam naravno nekoliko saveta za bezbedno sletanje bez kompjutera

Otvor kroz koji se ulazi u stanicu je uvek okrenut prema planeti Najlakše ćete ga pronaći ako se približavate stanici tako da vam ona cetaje se desna s planeta sa leve strane Gledajte kroz desni prozor sve dok na ugledate stanicu a zatim zakočite i okrenite brod prema njoj Dajte umereni gas i pokušajte da usaglasite rotaciju broda sa rotacijom stanice idealna je situacija u kojoj se vrata stanice prividno ne kreću Sam ulazak ne morate da izvodite minimalnom brzinom kao što piše u uputstvu za upotrebu programa uz dosta vežbe ćete čak naučiti da ulećete u stanicu punom brzinom Najvažnije je da uvežbate sletanje, nema nikakvog smisla da počinjete trgovinu ili gusarenje ako ne umete nepogrešivo da sletite i unovčite rezultate svog truda!

Galactic Hyperspace (Hiper pogon) je veoma skup i teško ga je nabaviti Omogućava vam da pređete u susednu galaksiju (galaksije su obeležene brojevima 1—8 iz devete ponovo dolazite u prvu) i tako posetite čitav niz novih svetova Skok u susednu galaksiju koji će potrošiti vaš Galactic Hyperspace se naročito ne isplati — tamo nećete naći ništa novo a i cene raznih roba se neće bitno razlikovati Jedina korist od intergalaktičkog skoka je što drugu misiju dobijate isključivo ako se nalazite u trećoj galaksiji

Za opremanje broda je dakle potrebno dosta novca koji se može zaraditi isključivo trgovinom (i gusarenje nije loš izvor prihoda ali za njega treba dobro opremiti brod) Mnogo ćete pogrešiti ako počnete da trgujete pošteno — čekovni račun će vam rasti sasvim postepeno dok će vas neprijatelji neprekidno napadati da bi vam preoteli teret Daleko je bolje odevijati manje ali unosnije poslove švercovati narkotika Narkotici se naime mogu kupiti za svega četiri kredita po kilogramu dok se na nekim drugim svetovima mogu prodati i za 104 Obzirom da kada kupite Large Cargo Bay' (Veliko spremište) možete poneti 35 tona narkotika u jednom putovanju, zarada će iznositi celih 3500 kredita! Nekoliko švercerskih podutivata i bićete dovoljno opremljeni da se upustite u jednu još nečasniju rabotu uništavanje policijskih brodova.

## Igra za nepoštene

Za ovaj je zločin potrebno posedovati prednje i zadnje military lasere Izletite iz stanice i sačekajte da se malo udaljite od nje a zatim potpuno zaustavite brod i aktivirajte zadnji prozor Otvorite snažnu vatru iz lasera na ulaz u svemirsku stanicu (gađajte je i projektilom) i za trenutak će se na izlazu pojaviti dva policijska broda. Neka vas to ne zabrinjava — nekoliko hitaca će ih uništiti a te će sudbina zadesiti i sve koji ih budu sledili Kada se vaš brod previše udalji od stanice brzo ga okrenite za 180 stepeni uništite policijske brodove koji su u međuvremenu izleteli i lagano se približavajte stanici uništavajući nove progonitelje Za to će vreme vaš policijski status postati Fugitive ali vas to neće previše brinuti — nekoliko skokova kroz hipersvemir i vaša će se potemnica izgubiti Važno je međutim da zapamtite dve stvari nikada ne pokušavajte da sletite na stanicu u čijoj ste blizini postali fugitive i nikada ne napadajte policijske brodove dok nosite narkotike Sve je prilika da će vas pratiti kordon policije ali je vrlo neverovatno da će vas napasti Ako se napad ipak desi borba vam ne koristi — čak i ako uništite sve progonitelje nećete moći da sletite na stanicu. Jednostavno skočite do susednog sistema kroz hipersvemir Uništavanje policijskih svemirskih lovaca vam nikada ne može doneti neku neposrednu korist, a neće vam biti ni zanimljivo ono je međutim, najbrži način da steknete viši rejting

Uništavanjem jednog broda (i postizanjem nekih drugih ciljeva) vaš se skor naime povećava za 1 Čim uništite 256 brodova računar će ispisati poruku 'Right On Commander' (Razumem komandante) i vaš će se rejting popeti za jednu stepenicu (rejtinzi su kao što verovatno znate Harmless (Bezopasan) Mostly Harmless (Uglavnom bezopasan) 'Poor (Slab) Average ( Prosečan), Above average (Iznadprosečan) Competent Sposoćen) Dangerous (Opasan) Deadly (Smrtonosan) i konačno, Elite (Elitan) U početku će svaka Right On Commander poruka povećati vaš rejting za jednu stepenicu ali će biti veoma teško preći poslednji korak — da biste se priključili 'Eliti treba da dobijete 25 poruka to jest da uništite oko 6500 brodova! Možete nam verovati na reč da nije ni malo lako postići ovaj skor Jedini je način da uništavate sve brodove na koje naiđete ne razmišljajući da li je reč s piratima ili o mirnim trgovcima Od starog kaubojskog principa „prvo pucaj pa onda postavljaj pitanja" treba odustati jedino u sigurnosnoj zoni svemirske stanice čak i ako vas gusari napadnu, bežite od njih ili skočite u susedni sistem U protivnom ćete sebi navući na vrat policiju i to u situaciji u kojoj niste spremni da joj se suprotstavite (čak i ako ste spremni okršaj sa policijom znači automatsku zabranu ulaska u stanicu)

slika 2

```
 10 REM
 20 REM          E L I T E
 30 REM
 40 REM  Kreiranje 'Harmless' ali
 50 REM   prababagateg  Komandanta
 60 REM
 70 REM   Dejan Ristanovic 1986
 80 REM
 90 REM        "Racunari 12"
100 REM
110 REM
120 c=0
130 FOR I=&4000 TO &404B
140     READ a
150     c=c+a
160     ?I=a
170 NEXT I
180 IF c<>3948 THEN
    PRINT "Proveri DATA liste":END
190 FOR I=&404C TO &404F
200     ?I=0
210 NEXT I
220 *SAVE E.Krez 4000 +100 0 0
230 END
240
250 REM Zamena linije 220 za
260 REM kasetnu verziju Elite:
270
280 *SAVE Krez 4000 +100 0 0
290
300 DATA 14,16,160,82,210,66,16,154
310 DATA 189,32,10,28,42,70,132,3
320 DATA 151,151,151,151,0,0,37,0
330 DATA 0,0,0,0,0,0,0,0
340 DATA 0,0,0,0,0,0,0,0
350 DATA 255,255,0,20,255,255,255,0
360 DATA 0,0,0,4,0,6,10,2
370 DATA 0,3,54,32,56,40,41,29
380 DATA 0,53,2,15,2,0,60,0
390 DATA 0,0,207,134
```

Ovaj napis završavamo programom sa slike 2. njegovim ćete startovanjem na kaseti ili disku dobiti status 'Harmless' komandanta čiji je brod opremljen kompletnim inventarom uključujući i strahovito snažnu energetsku jedinicu koja se ne može nabaviti ni na jednom svetu Dobar način da započnete igru zar ne? Ne pokušavajte ipak da sami menjate podatke o komandantu (promenom 72. bajta iz DATA liste u &20 postali biste na primer Elita") jer su programeri Acornsofta u program ugradili veoma složen čeksum koji otkriva sve manipulacije U idućim ćemo „Računarima" otkriti tajne ovoga čeksuma i objaviti program pomoću koga ćete lako i brzo kreirati komandante proizvoljnih karakteristika.

***Dejan Ristanović***

*Spektrumovci u nevolji*

# smrtonosna dijagnostika

## *Kad spektrum stane. . .*

Kada govorimo o otklanjanju kvarova, u hronično slabim uslovima postojanja i delovanja servisne mreže, pojavio se na tržištu priličan broj samostalnih poduzetnika koji nude usluge ovog tipa. Ti ljudi se bave popravljanjem na manje-više amaterskoj osnovi ali su obično pravi spas za vlasnika neispravnog spektruma.

### *Nakon kvara*

Servisere muči niz problema. Prvi i najveći je problem nabavke i visoka cena rezervnih delova. Centralni procesor i standardna logička kola uglavnom nisu problematični. Malo veća prepreka su 32K dinamički RAM-ovi. Bez podatka da su to u stvari pokvareni 64K RAM-ovi ne može se mnogo uraditi. ROM se može programirati ali je zato nepremostiv problem ULA. Za nju je još uvek jedini dostavljač Sinkler i njegovi distributeri.

Sledeći korak kod organizovanja servisa „u garaži" je nabavka opreme i mernih instrumenata. Dobra lemilica i manji osciloskop su na gornjoj granici dostižnog. Na tom nivou možete otklanjati jednostavnije kvarove dok je za komplikovanije potreban logički analizator ili originalna Sinklerova naprava za testiranje. I jedno i drugo je izvan domašaja našeg servisera.

### *Olakšana dijagnoza*

Osnovna ideja u ovom projektu je dijagnosticiranje pomoću jednostavnih programa koji se izvršavaju u centralnoj procesorskoj jedinici računara. Kada procesor izvršava samo jednu do dve instrukcije u petlji mogu se na osciloskopu lepo videti tokovi signala. Pri normalnom radu to nije moguće pošto CPU izvršava mnoštvo raznih instrukcija pa se signali ne ponavljaju.

Kada se jednom tok pojedinog signala uhvati na ekranu osciloskopa, nalaženje greške više nije problem.

Uređaj za testiranje se uključuje u pokvareni spektrum pomoću standardne utičnice za proširenje sistemskog basa. Kada je uređaj priključen spektrumov ROM sa sistemskim programom i bejzikom je ugašen a upaljen ROM u tekstu. Ovaj ROM sadrži zbirku programa za testiranje koji se biraju pomoću preklopnika na komandnoj ploči uređaja.

### *Konstrukcija testera*

Prototip uređaja za testiranje sadrži dva ROM-a, odnosno eproma — jedan sa programima za testiranje — IC7 i kopiju spektrumovog ROM-a — IC8. Trenutno aktivni ROM se bira preklopnikom na komandnoj ploči — S4. Ovim je izbegnuto višekratno isključivanje i uključivanje uređaja za testiranje kad god želimo da aktiviramo spektrumov sistemski program.

Adresne linije spektrumovog basa se vode neposredno na oba ROM-a. Eprom sa programima za testiranje dobija od CPU samo adrese od A0 do A7 a A8 do A11 se postavljaju pomoću preklopnika S0 S1 S2 S3. Na taj način se bira program za testiranje koji se izvršava nakon što procesor primi signal RESET.

Vidimo da eprom sa programima za testiranje sadrži najviše 16 programa, najveće dužine od 256 bajtova svaki.

Dvoja „NAND" vrata IC3/1 2 3 i IC3/4 5 6 su vezana kao multivibrator. Stanje multivibratora se bira pomoću dvostrukog trenutnog preklopnika S4. Izlazi TEST odnosno SPECTRUM odlučuju u kojem režimu uređaj radi. Svaki od signala omogućuje rad jednog od dva adresna dekodera. Integrisana kola IC4/9 10 11 13, 14 — IC3/8 9,10 — IC3/11 12 13 — IC6/1 2,4 5 6 signalom TEST=1 uključuju eprom sa programima za testiranje dok IC4/2 3 5 6 7 — IC6/8 9 10 12 13 signalom SPECTRUM=1 uključuju eprom sa sistemskim programom.

Treći adresni dekoder IC5/1 2 3 — IC5/4 5,6 omogućava programsko postavljanje stanja bistabilnih multivibratora IC2 kada je signal IZLAZ=0. Izlazna stanja se prenose na otpornike R1 do R8 ako je S6 vezan na masu. Logičko „1" na bilo kom od izlaza IC2 dovodi odgovarajući tranzistor T1 do T8 u stanje zasićenja, te kolektorskom strujom pali odgovarajuću svetleću diodu D1 do D8.

Izlazni sklop reaguje na izvršavanje OUT instrukcije u CPU sa adresom 1101 1111 binarno odnosno &DF. Stanje akumulatora se prenosi na IC2, odnosno priključenom S6 na LE diode.

Signal ROMCS na spektrumovom basu se postavlja na logički nivo „1" čime se „gasi" unutrašnji ROM a trenutno povezivanje linije RESET na masu preko S5 omogućava uspostavljanje početnog stanja u spektrumu.

IC1 sa filtrima C1 C2 C3 snabdeva logička kola uređaja električnom strujom za napajanje. Priključci za napajanje se nalaze se na svakom integrisanom kolu na istom mestu. Njihov raspored (brojevi nožica) dat je u sledećoj tabeli

```
*HISOFT GENS3M2 ASSEMBLER*
      ZX SPECTRUM

Copyright (C) HISOFT 1983,4
All rights reserved

Pass 1 errors: 00

             1 *C-
00DF        10 PORT    EQU  #DF
            20
C000        30         ORG  #C000
            40
C000        50 LOOP    JR   LOOP
            60
C100        70         ORG  #C100
            80
C100        90         LD   D,#00
C102       100         LD   C,#00
C104       110 LOOP_1  LD   HL,#4000
C107       120 LOAD_1  LD   A,L
C108       130         XOR  H
C109       140         XOR  D
C10A       150         LD   (HL),A
C10B       160         INC  HL
C10C       170         LD   A,H
C10D       180         CP   #80
C10F       190         JR   NZ,LOAD_1
C111       200         LD   HL,#4000
C114       210 READ_1  LD   A,L
C115       220         XOR  H
C116       230         XOR  D
C117       240         XOR  (HL)
C118       250         OR   C
C119       260         LD   C,A
C11A       270         OUT  (PORT),A
C11C       280         INC  HL
C11D       290         LD   A,H
C11E       300         CP   #80
C120       310         JR   NZ,READ_1
C122       320         INC  D
C123       330         JR   LOOP_1
           340
C200       350         ORG  #C200
           360
C200       370         LD   D,#00
C202       380         LD   C,#00
C204       390 LOOP_2  LD   HL,#8000
C207       400 LOAD_2  LD   A,L
C208       410         XOR  H
C209       420         XOR  D
C20A       430         LD   (HL),A
C20B       440         INC  HL
C20C       450         LD   A,H
C20D       460         CP   #00
```

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| +5 | IC1/3 | IC2/20 | IC3/14 | IC4/16 | IC5/14 | IC6/14 | IC7/24 | IC8/28 |
| GND | IC1/2 | IC2/10 | IC3/7 | IC4/8 | IC5/7 | IC6/7 | IC7/12 | IC8/14 |

### *Programska oprema*

Program za testiranje dat je u asembleru na listingu 1. Startovanjem programa u računaru će biti proizveden odgovarajući kod koji treba presnimati u eprom. Osim tri kratke rutine, sve ostalo će biti bajtovi & 76.

Programiranje uređaja za testiranje u okvirima datih mogućnosti hardvera ograničeno je jedino maštom konstruktora. Vodeći princip je prilagođenje programa tipu greške koja se dijagnosticira te koncipiranje testova od jednostavnijih ka složenijim.

Pogledajmo kako su ta načela uzeta u obzir u našem prototipu.

Na početnoj adresi prvog testa 0000 nalazi se skok na samog sebe. To je najjednostavniji pokušaj da se ustanovi da li uopšte rade CPU i bas. Ukoliko pokušaj ne uspe CPU izleće u HALT instrukciju (76) procesor se zaustavlja i — greška će brzo biti locirana!

Sledeći testovi se bave memorijskim bankama. Na & 0100 je test za memorijsku banku na adresi &4000. Taj RAM imaju svi spektrumi, a koriste se za sistemske varijable, ekran i korisnikov program. Memorija se prvo puni određenim sadržajem a onda čita i proverava. U slučaju greške pali se jedna od svetlećih diode na izlaznim vratima & DF. Redni broj LED diode označava koji od RAM čipova je u kvaru.

Na adresi & 0200 nalazi se sličan program za proveru delovanja memorijske banke na adresi & 8000. Ovu banku imaju

***Spektrum, u tehnološkom smislu, spada među najslabije mašine na tržištu. Zbog komercijalnih razloga, on je izrađen prilično amaterski — kako konstrukcijom tako i pouzdanošću rada, on spada u donji razred klase „zabavne elektronike''. U doba uvođenja u prodaju, to mu je, između ostalog, dalo odlučujuću prednost pred konkurentskim proizvodima, dok danas još samo izaziva glavobolju velikom broju vlasnika. Ali, šta je tu je: oni imaju spektrum moraju da žive s njim i — stalnim strahom od iznenadnog kvara! Stoga uređaj za dijagnostiku kvarova nije namenjen samo potencijalnim serviserima, nego i nepreglednoj armiji uradi-sam amatera kojima će se kad-tad neko od rodbine ili prijatelja obratiti za pomoć.***

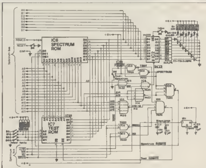
```
C20F   470          JR   NZ LOAD_2
C211   480          LD   HL,#8000
C214   490 READ_2   LD   A,L
C215   500          XOR  H
C216   510          XOR  E
C217   520          XOR  (HL)
C218   530          OR   C
C219   540          LD   C,A
C21A   550          OUT  (PORT),A
C21C   560          INC  HL
C21D   570          LD   A,H
C21E   580          CP   #00
C220   590          JR   NZ,READ_2
C222   600          INC  E
C223   610          JR   LOOP_2
       620
C300   630          ORG  #C300
       640
C300   650          LD   C,#00
C302   660 SET_ON   LD   HL,#0000
C305   670 WAIT     INC  HL
C306   680          LD   A,H
C307   690          OR   L
C308   700          JR   NZ,WAIT
C30A   710          INC  C
C30B   720          LD   A,C
C30C   730          OUT  (PORT),A
C30E   740          JR   SET_ON
       750
E000   760          ORG  #E000
       770
E000   780 HALTS    LD   HL,#C002
E003   790          LD   DE,#C003
E006   800          LD   BC,#0FF-#002
E009   810          LD   (HL),#76
E00B   820          LDIR
E00D   830          LD   HL,#C125
E010   840          LD   DE,#C126
E013   850          LD   BC,#1FF-#125
E016   860          LD   (HL),#76
E018   870          LDIR
E01A   880          LD   HL,#C225
E01D   890          LD   DE,#C226
E020   900          LD   BC,#2FF-#225
E023   910          LD   (HL),#76
E025   920          LDIR
E027   930          LD   HL,#C310
E02A   940          LD   DE,#C311
E02D   950          LD   BC,#FFF-#310
E030   960          LD   (HL),#76
E032   970          LDIR
E034   980          RET
       990
E000  1000          ENT  HALTS

Pass 2 errors: 00

Table used:   149  from   800
Executes: 57344
```

## Program TEST

*Ovaj program će generisati kompletan kôd (4096 bajtova) spreman za unošenje u EPROM. Treba resetovati računar i učitati DEVPAC GENS asembler počev od adrese 30000. Posle startovanja asemblera, ukucati ceo program i asemblirati ga. Zatim izvršiti naredbu R koja će startovati program od adrese HALTS (&E000). Time se generišu sve potrebne HALT instrukcije. Po povratku u bazik formirani kôd se može snimiti na kasetu ili se direktno preneti u EPROM. Početak bloka je na adresi &C000 (49152) a dužina 4096.*




### Spisak sastavnih delova

| Kod | Broj | Elementi | Dodatni podaci |
|---|---|---|---|
| IC1 | 1 | 7805 | Stabilizator napona 5V/1A, kućište T0220 |
| IC2 | 1 | 74373 | TTL osmostruki latch, kućište DIL20 |
| IC3 | 1 | 7400 | TTL četvorostruki NAND sa po 2 ulaza, kućište DIL14 |
| IC4 | 1 | 7423 | TTL dvostruki NOR sa po 4 ulaza, kućište DIL16 |
| IC5 | 1 | 7432 | TTL četvorostruki NOR sa po 2 ulaza, kućište DIL14 |
| IC6 | 1 | 7420 | TTL dvostruki NAND sa po 4 ulaza, kućište DIL14 |
| IC7 | 1 | 2732 | EPROM 4K×8/450nsek sa programom za testiranje |
| IC8 | 1 | | Spectrum ROM (Nije nužno potreban) |
| T1-T8 | 8 | | NPN Tranzistor |
| D1-D8 | 8 | | Svetleća LED dioda, boja crvena |
| C1 | 1 | | Keramički kondenzator 0,1µF/50V |
| C2 | 1 | | Elektrolitski kondenzator 100µF/25V |
| C3 | 1 | | Tantalski kondenzator 1µF/15V |
| R1-R8 | 8 | | Karbonski otpornik 4,7K/0,25W |
| R9, R10 | 2 | | Karbonski otpornik 1K/0,25W |
| R11-R18 | 8 | | Karbonski otpornik 330E/0,25W |
| R19-R22 | 4 | | Karbonski otpornik 1K/0,25W |
| S0-S6 | 7 | | Mikroprekidačnik DPDT |
| | 1 | | Prototipna štampana pločica 100×160 |
| | 1 | | Konektor EDGEBOARD (28+28)×0,1 |

semo spektrumu 48K a koristi se isključivo za korisnikove programe.

Na adresi &0300 se nalazi program koji na svetlećim diodama izvodi mali light show.

* * *

Uz utrošak srazmerno skromnih sredstava može se izraditi vrlo snažno pomagalo za otkrivanje grešaka i kvarova na spektrumu. Za rad sa uređajem nije potrebno podrobno poznavanje centralne procesorske jedinice i ostalih logičkih kola. Kod upotrebe logičkog analizatora koji košta nekoliko desetina pa i stotina puta više serviseri i konstruktori mogu biti dobro potkovani i ovakvom vrstom znanja.

***Štefan Kirn, Karel Svetek, Davor Vilčić***

U SVETU KOMPONENTI

# Šta ima novo

## Inteligentni displej

*Američka firma Litronix (sada u sastavu Siemens-a) nudi veliki izbor inteligantnih višecifarskih LED pokazivača koji se lako povezuju sa postojećim mikroprocesorima CPU ih vidi kao RAM) pošto sadrža i intarni generator ASCII karaktera mogućnost upravljanja kursorom, radni napon je standardni 5V Oznaka ovih komponanti koje su idealna da se sa njima napravi inteligentni DISPLAY su DL 1416 DL 2416 DL 3416, DL 1814 (osam ASCII znakova)

## Prvi u Evropi

Siemens je prva evropska firma koja je započela proizvodnju dinamičkih memorija tipa RAM kapacitata 256 K (postoje dva modela sa po tri brzinska varzije vremena pristupa su 120/150/200 ns)

## Ka superkompjuteru

Ako se spremate da pravite dobar računar, morate znati gde su vam (komercijalna) granice mikroprocesori se prave do 32 bita UV EPROMi sa kapacitetom do 512 Kbita (planira se 1 Mbit) CMOS EPROMI do 256 Kbita ROM-ovi (Factory Masked) do 1 Mbit dinamička memorija do 256 Kbita (uskoro 1 Mbit) statička RAM memorija (isključivo CMOS) do 64 Kbita pseudostatička (periferija CMOS a interna dinamička matrica HMOS) do 256 K Floppy diskovi od 5 25 inča do 1 6 Mb Winchester diskovi od 5 25 inča do 280 MB SMD diskovi reda gigabajte itd itd Naravno pod „komercijalno nabavljivim" komponentama podrazumevamo ono što se danas može kupiti na tržištu a ne ono što se najavljuje u reklamama ili je još uvek u razvoju

## Mini vinčester

Već danas se prave „mali" Winchester diskovi (3 5 inča) sa kapacitetom od oko 13 megabajta (rodime)

## RAM . . .

Ako ste se već odlučili da napravite radnu memoriju sa RAM dinamičkog tipa kapacitata 256 K onda bar iskoristite 8419 DRAM kontrolar (National Semiconductor) koji obavlja celokupan „posao" bez usporavanja do CLOCK-a od 16 MHz (bez WAIT stanja) i lako se povezuje sa praktično svim CPU (mada najbolje radi sa 32032) Primenljiva DRAM su sa pristupom od 150 ns ili sporije a ako se ovom kontroleru pridoda DP 8400-2 ECC Unit (Error Checking and Correction) onda ovaj set garantuje uvek tačne podatke iz memorije

## . . i disk kontroleri

Do pre nekoliko godina kontrolari iz flopi disk jedinice su bili građeni kao štampana ploča sa preko dve stotina TTL (SSI i MSI) integrisanih kola danas se takav sklop pravi sa jednim MOS LSI kolom (Floppy Disk Controller — FDC) i nekoliko TTL bafera Najveći izbor FDC LSI kola ima američka firma Western Digital (po licenci ih pravi i Siemens) a jedan od najkompleksnijih među njima je FDC sa oznakom WD 2797 koji podržava sve vrste diskova PLL data separator prekompenzacija upisa standardna veza (interface) kako prema disk jedinici tako i CPU basu transfer podataka iz/u memoriju ide direktno (DMA) po sektorima a mikroprocesor sa njim komunicira lako razumljivim makroinstrukcijama

## Statičke memorije

Statička RAM su dostigla zavidnu popularnost u mikroračunarima zahvaljujući pre svega dobrim osobinama prvih modela na tržištu Hitachi serije HM6116 od 16 Kbita (2 K×8) kakva se koristi i u galaksiji a zatim i HM6264 od 64 Kbita kao što su solidna brzina (reda 100 ns) i niska potrošnja (aktivna struja 40 mA „stand-by" potrošnja reda 100 uA) Iako ih Hitachi u katalogu predstavlja kao CMOS statička RAM, one su napravljene tzv „MIXED CMOS" tehnikom (što znači NMOS matrica kao „srce" memorije a CMOS periferija), dok Toshiba pravi slična tipova statičkih RAM (to im je specijalnost) čistom CMOS tehnologijom, što je skuplje ali daje bolje rezultate uz istu brzinu aktivna struja je reda 5 mA, a „stand-by" potrošnja samo 0 2 uA (max!) Obe firme su najavile i 256 K tipove (32 K×8) ali im je pristup drugačiji (Hitachi pravi pseudostatički RAM)

## Prekidački regulatori

Blok napajanja u malim personalnim računarima je obično izveden kao klasičan ispravljač plus linearni regulator za 5 V (7805 u „galaksiji" ili spektrumu je dobar primer fiksnog naponskog regulatora) Međutim kako su lični računari dobijali u kompleksnosti (potrošnja energije raste) tako su i ispravljači/regulatori postajali sve veći (veća disipacija veći hladnjaci veća težina manja pouzdanost) što je neminovno uslovilo primenu prekidačkih izvora napajanja tzv „Switching Mode Power Supply — SMPS") koji su „do juče" imali svoja dečja bolesti Razvojem modernih prekidačkih komponenti POWER MOSFETa Feriti kondenzatori sa malim ESR/ESL IC za pobudu i regulaciju SMPS) ovi izvori napajanja naglo zamenjuju linearne regulatore zbog mnogo veće efikasnosti (znači mala disipacija mali hladnjaci mala težina i zbog feritnih transformatora) malih dimenzija (posledice rada na nekoliko stotina kHz) i veće pouzdanosti

Ako vam je potreban izvor napajanja za 5 V i nekoliko ampera (2—3), možete izabrati između monolitnog regulatora firme SGS (ista je inače poznata po svojim POWER komponentama) tipa GS-R400 koji daje 5 V pri 4 A uz skoro 90% efikasnosti ima svoj mali hladnjak dodatne izlaze (npr reset), tačnost 1% pri ulaznim naponima od 12—30 Volti i Hibridnog SMPS regulatora firme RIFA (PKA serija) koji pored 5 V daje i druge napone (npr +12 V za serijski I/F) ima snagu od 40 W uz efikasnost od 85% radi na 300 kHz dimenzija reda 9×9×2 cm, regulacija izlaznog napona je 10 mV i ima srednje vreme između otkaza (MTBF) od 1 7 miliona časova (preko 200 godi) zahvaljujući činjenici da nema ugrađenih elektrolita Ove POWER komponente su napravljene za princip „ugradi-pa-zaboravi" a njihove mane (koje postoje) nisu bitne za digitalne sisteme Linearni regulatori će i dalje imati primenu u analognim kolima zbog bržeg odziva i manje talasnosti U računar BBC inače odavno se ugrađuje prekidački regulator

***Priprema: Miše Blažimir, dipl. ing.***

# Mali oglasi

*Ako ne možete da podnesete da drugi nemaju ono što vi imate objavite svoj mali oglas u „Računarima"*

*Ako ne možete da podnesete da drugi imaju ono što vi nemate javite se na neki od malih oglasa u Računarima'*

*Ako ne volite da da se dopisujete sa „Računarima" svoj mali oglas možete nam izdiktirati preko telefona 011/650-161 svakog radnog dana od 10—14 sati Mi ćemo vam onda naknadno poslati ispunjenu uplatnicu*

*Prva stvar koju treba da uredite je da se odlučite da li želite običan ili uokviren mali oglas*

*CENA OBIČNOG MALOG OGLASA do dvadeset reči je 600 dinara Svaka naredna reč košta još 40 dinara s tim što oglas ne sme da ima više od 50 reči Adresa oglašivača na računa se u cenu*

*CENA UOKVIRENOG MALOG OGLASA je 600 dinara po visinskom centimetru, s tim što se mogu zakupiti najmanje po 32 slovna znaka Ako se ne iskoristi čitav prostor u jednom redu, računa se broj redova a ne broj znakova Za uokvirena oglase preko 5 cm cena je 900 dinara po centimetru*

*Poželjno je da vaš mali oglas počinje sa Prodajem, Kupujem Držim časove, Menjam ili načim sličnim što ukazuje na sadržaj oglasa*

*Svi mali oglasi koji stignu u našu redakciju do 28 februara do 12 sati biće objavljeni u „Računarima" broj 13 koji izlaze iz štampe polovinom marta*

*Da ne bi bilo zabune obavezno naznačite da li želite običan ili uokviren mali oglas, i zajedno sa tekstom vašeg malog oglasa pošaljite i priznanicu o uplati na adresu redakcija „GALAKSIJA", BULEVAR VOJVODE MIŠIĆA 17, BEOGRAD sa naznakom „za male oglase u Računarima"*

## SPEKTRUM

SPEKTRUMOVCI HOBBY SOFT opet nudi hit programa po niskim cenama Usluga brza i kvalitetna. Snimanje na PANASONIC kasetofonu za kompjuter Besplatan mali katalog Đakovački Igor Karlovečka 18 11080 Zemun tel 011/219-702 Nećamo vas iznevariti!

• ANALOG 20 — za sve koji žele upoznati svijet analognih kompjutera na Spektrumu Brzo i stručno rešavanje svih dinamičkih sistema iz oblasti strojarstva elektrotehnike robotike NC upravljenja regulacija nelinearnih diferencijalnih jednačina operacionog istraživanja Programski pakat za stručnjake i one koji to žele postati LAPLACE CONTROL SOFT Cesta na Svetino 19 63270 Laško

□ Spektrumovci — OFB COMPUTER STUDIO vam nudi komplet od 10 programa po cijeni od 700 din +(400 din kaseta i poštarina) Komplet 1 sadrži programa kao npr SNIPI GLASS SOHO BEX OUEST W S BASKETBALL itd Prva dva kupca od 20 3 1986 plaćaju samo kasetu i poštarinu Omar Eškić Fra Grge Martića 31 78000 Banjaluka tel 078/31-422

□ Prodajem Spektrum Plus ocarinjen interfejs džojstik literaturu i preko 200 programa Milanović Vladimir Kristijana Karpoša 43 92000 Štip tel 092/21-232

• BLAST — ubrzajte svoja bejzik programa do 40 puta upotrebom do sada najboljeg kompajlera za Spektrum Trake sa BLAST COMPILER-om i BLAST TOOLKIT-om i opširno uputstvo — 1500 din MACHINE CODE FOR BEGINNERS — uživajte u efikasnosti mašinskog programiranja Trake i opširno uputstvo — 1500 din Oba programa sa uputstvima — 2500 din VRCA Milan Zarija Vujoševića 79 11070 Novi Beograd

□ Spektrumovci! Prodajem programa po najnižim cenama Tražite besplatan ilustrovani katalog IMPOSSIBLE MISSION WINTER GAMES F WARRIOR POPEYE MARSPORT ELITE Tadić Slobodan Rumenački put 5 21000 Novi Sad tel 021/317-910 ili 311-576

• Spektrumovci! MERLIN SOFT — najbolja YU piratska kompanija objavljuje sletanja svog specijalnog aviona na beogradski aerodrom na liniji London—Beograd U avionu su najpoznatiji svetski soft hitovi Pentagram, Myra Mara Mugsy's Revenge Superman Critical Mass Yie Ar Kung Fu Transformers Winter Games Shadow of Unicorn Fairlight Petrović Aleksandar Olge Alkalaj 7/113 11000 Beograd tel 011/4880419

• Verovali ili ne GHOSTBUSTERS STUDIO prodaja komplete najnovijih programa za ZX Spektrum po ceni od 800 din Babić Srđan Generala Ždanova 48 11000 Beograd tel 658-031

• Spektrumovci — NEEDLE — program za presnimavanje (snima 99% programa) Poseduje opciju za ubacivanje POKE-ova direktno u program prilikom presnimavanja — pouzdano Program + uputstvo + 100 besmrtnosti = 750 din (Na vašu kazetu sve sa 490 din) Pouzećem Branko pp 57 47300 Ogulin tel 047/72-289

□ Prodajem Spectrum 48K mikrodrajv ZX interfejs printer — pisaću mašinu Brother EP-22 sve potrebne kablove papir i trake za printer programe Ćirić Zoran Splitska 4, 11000 Beograd tel 011/414-274

• 12 SUPER HITOVA za Spektrum samo 700 din + kaseta D T Supertest 1 i 2 Dambusters, Exploding Fist Nodes of Yesod Boulder Dash 2 Macadam Bumper Impossible Mission Fighting Warrior Sorcery International Karate Skool Daze 2 ASTERIX Club Ljube Nenadovića 1/8 11500 Obrenovac Nikolić Dušan tel 011/873-777

□ SPEKTRUMOVCI trenutno najbolji programi pojedinačno i u kompletima Komplet 8 BACK TO SKOOL SUPER PIPELINE 2 BEACH HEAD 2 FIGHTING WARRIOR W S BASKETBALL POPEYE COMMANDO RASPUTIN SABOTEUR McADAMB BUMPER BOUNTY BOB ROLLER COASTER Komplet 7 SKOOL DAZE 2 BEACH HEAD 2 BOULDER DASH 2 INTERNATIONAL KARATE 1 INTERNATIONAL KARATE 2 IMPOSSIBLE MISSION SORDERONS SHADOW DYNAMITE DAN MONTY ON RUN EXPLOADING FIST OUNDARACH FRENKIE Jedan komplet + kaseta + poštarina 1400 — din Nestorović Srđan Oslobođenja II deo 8 11194 Beograd — Rušanj tel 011/888-222

• MOBIL SOFTWARE — najnoviji hitovi za Spektrum (Elite, Winter Sports Commando) i mnogo drugih Cena pojedinačno 50 d u kompletu 25 programa za 1000 d Katalog besplatan Gojković Dragomir Ljermontove 2/151 11104 Beograd tel 011/4681-758

□ Uporabne programe za spektrum prodajamo Izrađujemo programe po narudžbi Možete nabaviti mnogo novih igara (SEX MISSION 2 + kazeta 1000 din ) Tražite katalog Bralih Andraj Titova 18 64270 Jesenice ili Brišar Lado Tavčarjeva 3b Jesenice

□ SPEKTRUMOVCI već tri godina stojimo vam na usluzi svojom visokom profesionalnošću i najvećim mogućim izborom starijih i najnovijih programa Garantiramo kvalitetu i ekspresnu isporuku Naručite katalog (50 d) sa oznakom za Spectrum Prodajamo komplete po dvanaest programa (1000 dinara plus kaseta) uz velike popuste za veće narudžbe Za odrasle poseban program Sex Mission (sa uputama kazetom i troškovima otpreme 950 d ) FUTURE ORION 41000 ZAGREB Rubetićeva 7 tel 041/417-052

□ Prodajem mikrodrajv i interfejs 1 za spektrum Tel 011/214-860

□ SPEKTRUM — NAJNOVIJI I NAJBOLJI PROGRAMI U KOMPLETIMA KOMPLET 49 YIE AR KUNG FU ELITE RAMBO ROBIN OF WOOD FAIRLIGHT MIKIE SIR FRED TALOS WRIGGLER CLUEDO TRANSFORMERS SUPER BRAT KOMPLET 50 B C S QUEST FOR TIRES CRITICAL MASS CYLON ATTACK ROLLER COASTER JETSET WILLY 3 CHAOS NIFTY LIFTY ITD JEDAN KOMPLET + KASETA + POŠTARINA 1250 d KATALOG BESPLATAN TRTICA GORAN STEVANA LUKOVIĆA 9 11090 BEOGRAD TEL 011/563-348

□ Prodajem nov nekorišćen lajt pen za spektrum Lajt pen + uputstvo + kaseta pokion program + poštarina 9000 din Tel 022/413-342

• SPEKTRUM — PROFESIONALNI PREVODI NAPREDNI MAŠINSKI JEZIK 1500 d SPECTRUM ROM DISASSEMBLY 1500 d MAŠINSKI JEZIK ZA APSOLUTNE POČETNIKE 1300 d SPEKTRUM PRIRUČNIK 800 d MEGA BASIC UPUTSTVO 500 d MEGA BASIC NA KASETI 500 d DEVPAC 3 UPUTSTVO 600 d DEVPAC 3 NA KASETI 500 d 50 TAJNI BASIC PROGRAMIRANJA 500 d BETA BASIC 1 8 UPUTSTVO 400 d BETA BASIC 1 8 NA KASETI 500 d ARTIST UPUTSTVO 400 d ARTIST NA KASETI 500 d TRAŽITE BESPLATAN KATALOG TRTICA GORAN STEVANA LUKOVIĆA 9 11090 BEOGRAD TEL 011/563-348

□ Menjam i prodajem programe za spektrum Komplet od 14—48 programa samo 500 din Katalog 50 din Star War Soft Mariborska 123 35000 Svetozarevo tel 035/26-585

• SPEKTRUM — NAJNOVIJI I NAJBOLJI PROGRAMI U KOMPLETIMA KOMPLET A SKY RANGER SABOTEUR BACK TO SKOOL RASPUTIN I OF THE MASK NEVER ENDING STORY BORD ERON S SHADOW NACKER RATS
KOMPLET B FIGHTING WARRIOR BEACH HEAD 2 BOUNTY BOB STRIKES BACK IMPOSSIBLE MISSION DYNAMITE DAN BOULDER DASH 2 GLASS SUPER GRAN MACADAM BUMPER SORCERY INTERNATIONAL KARATE 1 i 2
KOMPLET C POPEYE W S BASKETBALL SUPER TEST 1 I 2 MONTY ON THE RUN MARSPORT SUPER PIPELINE 2 ACTION BIKER DAM BUSTERS TWO GUN TURTLE THAT'S THE SPIRIT ABU SIMBEL JEDAN KOMPLET + KASETA + POŠTARINA 1250 d TRAŽITE BESPLATAN KATALOG SA 1000 PROGRAMA TRTICA GORAN STEVANA LUKOVIĆA 9 11090 BEOGRAD TEL 011/563-348

□ Rečnik englesko-srpskohrvatski (oko 1400 najčešće upotrebljavanih reči-) + kaseta + poštarina (1300 din ) Tel 011/ /487-862 od 17—19 h

## SPEKTRUM




Sva za ZX SPEKTRUM
— video igra
— namanski programi
— programske uputstva
— literatura
— SPECIJALNO —
BETA BASIC 3 0 sa komplainim originalnim uputstvom kasalom i pošterinom za samo 1000 -d

Za sva usluga garancija! MILOVANOVIĆ LJUBIŠA Petra Lakovića 57 11030 BEOGRAD tel 011/558-007 posle 17 časova

• SPEKTRUM — 12 PROGRAMA ZA UČENJE ENGLESKOG + KASETA 1000 d 25 RADIOAMATERSKIH PROGRAMA + KASETA 1000 d 38 COPY PROGRAMA + KASETA 1000 d TRAŽITE BESPLATAN KATALOG SA NAJNOVIJIM PROGRAMIMA TRTICA GORAN STEVANA LUKOVIĆA 9 11000 BEOGRAD TEL 011/563-348

SPEKTRUM — VELIKI IZBOR NOVIH PROGRAMA IZ LONDONA POJEDINAČNO I U KOMPLETIMA PO POVOLJNIM CENAMA SABOTEUR BACK TO SKOOL I OF THE MASK TRAŽITE BESPLATAN KATALOG SA NOVIM PROGRAMIMA RANKOVIĆ MIRAN BRAĆE MIHAJLOVIĆA 46 11273 BEOGRAD

□ VRHUNSKI GRAĐEVINSKI PROGRAMI za ZX Spektrum OKVIRI REŠETKE, ROŠTILJI PILOTI DIMENZIONIRANJE VODOVOD i mnogi drugi Za radna organizacije i pojedince Besplatan katalog Gino Gracin Kozala 17 51000 RIJEKA tel 051/517-291

□ 5 — SOFT nudi vam komplete programa za spektrum Komplet sa vašom kasetom 700 din sa mojom 1000 din Katalog 100 din Milan Vunjak Mrakovička 50/8 11000 Beograd tel 011/535 532

□ SPEKTRUMOVCI — COMMANDO RASPUTIN SABOTER ROBBIN HOOD MARSPORT ELITE WINTER GAMES ARTIST TURBO po ceni od 60 din Svaka narudžbina 200 POKE-ova i 3 programa na poklon Besplatan katalog Saša Labudović Jovana Tomaševića 16 85000 Bar tel 085/27-616

□ Prodajam programe za Spektrum pojedinačno ELITE QUEST FOR TIRES RAMBO 2 ROBIN HOOD WRIGGLER ENIGMA FORCE CRITICAL MASS COMMANDO S H POP QUIZ Tel 021/716-024 Kisić Zoran Dr Mladena Stojanovića 8 21201 Rumenka

# OBRADUJTE SVOJ RAČUNAR! OBJAVITE MU MALI OGLAS U „RAČUNARIMA"!

SUPER TURBO II ubrzava učitavanje — snimanje (Program kaseta uputstva 1000 din ) Englesko srpskohrvatski rečnik sa 2000 reči (program kaseta uputstvo 1000 din ) Hit u Engleskoj — program ELITE sada i za Spektrum Cena programa 400 din + kaseta Kratič Aleksandar Balzakova 30 21000 Novi Sad

Prodajam programe za spektrum Komplet od 14 programa 450 din + kaseta Sa mojom kasetom 850 din Stojanović Vojkan Maršala Tita 42 — 2/4 35230 Ćuprija tel 035/64-777 ili 27-979

SPEKTRUMOVCI!!! KOMPLET 14 Winter Sports (8 disciplina) Commando School Battle Kung Fu 2 Roller Coaster Island Mopp Matagalctic Llamas Special Delivery Crazy Caverns Lancer Lord KOMPLET 15 Rasputin Robbin Hood Boulder Dash 2 Rambo B C Quest for Tires Critical Mass 10 Little Indians Mardons Quest Mary Xmas I of the Mask Sky Ranger S Strukar Cena jednog kompleta 800 din Radović Branislav Sonja Marinković 14/4 21000 Novi Sad tel 021/28-682 ili 022/424-824 (vikendom)

□ SPEKTRUMOVCI SUPERKOMPLET 16 Elite Enigma Force (Shadowfire 2 Wriggler Stripppokar 2 Uncle Tom Sapphire Transformers Mickie Hyperblaster Samuray Warrior Gyroscope Pacific War Cena 000 din Radović Branislav Sonja Marinković 14/4 21000 Novi Sad tel 021/28-682 ili 022/424-824 (vikendom)

□ SPECTRUM RAINBOW SOFTWARE vam nudi izbor od 70 raznih kompleta Kompleti od 25 programa 600 din Imamo i sva najnovija programa koji su trenutno već u Jugoslaviji Uverite se Katalog je besplatan Mihajlovski Kirčo Moša Pijada 128 91300 Kumanovo tel 0901/23-900

Prodajam kvalitetno snimljene programe za Spektrum Cena bilo kojeg programa je samo 50 dinara Stigli su svi novi programi koje ste čitali u drugim oglasima Janković Silard Petef Šandora 64 25222

## KOMODOR

KOMODORCI najpopularnije igre januar 86 GHOSTBUSTERS 2 JUNGLE HUNT 2 SWS 3 BOULDER DASH 3 ZORRO + kaseta 1200 din Može i pojedinačno Patković Dejan Milana Rakića 20 11000 Beograd tel 011/424-744

□ COMMODORE 20 16 +4 128 64 programi Tražite besplatan katalog Darman Šandor Rade Končara 23 23000 Zrenjanin

□ Za C-64 veliki izbor najnovijih programa ZORRO WAR GAMES FISTFULL OF BOOKS DYNAMITE DAN BOULDER DASH 3 i drugo Besplatan katalog Todorović tel 011/458-134

Novo za Komodor 64!
Prerađujem programe —
— Program koji čekate dve minute možete loadirati za svega 30 sekundi Prerađeni program ima auto-start
— Na disketu možete staviti tri puta više programa
— Prerađen program nema zaštitu
— Besplatno dobijate FILE COPY kojim kopirate prerađene programe
— Ušteda 1 u vremenu loadiranja.
— Ušteda 2 potrebno vam je tri puta manje disketa
Ovaj oglas je za vas
— ako radite programe
— ako skidate zaštite
— ako prodajete programe
— ako nemate mnogo disketa
— ako se igrate
— ako menjate programe
(Sve ovo samo na disketi)
Damir Brkić V Gaćeše 6 44260 Glina tel 044/88-221 od 8—12 i 16—18 h

□ Komodor 64 — izbor najboljih programa u kompletima od 20 komada (1000 din sa kasetom) ili pojedinačno po 100 din Besplatan Katalog Panić Saše Kumodraška nova 13 11000 Beograd tel 011/461-779

• COMMODORE 64 — PROFESIONALNI PREVODI REFERENCE GUIDE 1700 d MAŠINSKI JEZIK ZA POČETNIKE 1400 d PRIRUČNIK OD C-64 1800 d KAKO DA PROGRAMIRATE C-64 900 d DISK SISTEMI I ŠTAMPAČI 900 d GRAFIKA I ZVUK 800 d MATEMATIKA NA C-64 1000 d SIMON C BASIC 700 d C-64 BASIC 700 d PRACTCALC 800 d EASY SCRIPT 400 d VIZAWRITE 400 d PASCAL 400 d MULTIDATA 400 d HELP 400 d GRAF 400 d TRTICA GORAN STEVANA LUKOVIĆA 9 11090 BEOGRAD TEL 011/563-348

□ Prodajam 50 najboljih igara za COMMODORE 64 za 1000 dinara plus kaseta 400 dinara Kvalitet i usluga zagarantovani Isporuka odmah Stošić Slobodan 7 Juli 31 b/2 18000 Niš tel 018/51-805

• COMMODORE 64 — najnovija i najbolja programa za Vaš C-64 prodajamo Preko 3000 naslova Besplatan spisak DENI — OZREN DUKIĆ 41020 Zagreb Čalogovićeva 5/III tel 041/668-004

□ COMMODORE 64 — Paket igara Beach Head 2 Friday 13 F—15 Strike Eagle Exploding Fist Rambo 2 Winter Games 911 Tiger Shark Elite Staff of Karnath I Staff of Karnath 2 Commando prodajam Sa kasetom i poštarinom 1500 din Huselović Elmir A Herljevića 37 75000 Tuzla tel 075/216-044

□ Za COMMODORE 64 po ceni od 1500 din prodajam PIZZA DISK III koja kopira 225 blokova Ovim programom možete kopirati sa kasete na disketu DIGITAL DRUMS BLAGGER GOES T H ? ROLAND S RAT RACE i st KRESOJA DRAGIŠA VOJVODE STEPE 421/B 11000 BEOGRAD tel 011/475-418

□ Za COMMODORE 64 prodajam Komplet 1 SUMMER GAMES RAMBO II MC GUILLAN BOXING Komplet 2 AMAZON WARRIOR BLAGGER GOES T H DIGITAL DRUMS HACKER KUNG FU MASTER MACBETH Komplet 3 CHIMERA COMMANDO CYLU SKOOL DAZE SUPER HUEYE (4 helikopterske igre) Komplet 4 EMERALD ISLE ENTOMBED (STAF 2) FRANK BRUNO S BOXING SENTINEL SPITFIRE 40 VALHALLA itd Cena jednog kompleta je 500 din KRESOJA DRAGIŠA VOJVODE STEPE 421/B 11000 BEOGRAD tel 011/475-419

□ COMMODORE 64 — PYJAMARAMA 1 2 i 3 ELITE ACTION BIKER STOP EXPRESS JUMP JET BOULDER RASH II KNOCKOUT II i mnogi drugi po najnižim cenama Besplatan katalog Sijapčević Predrag Bulevar Revolucije 290/V 11000 Beograd tel 011/417-064

COMMODORE 64 — neverovatan izbor kasetnih programa 20 najboljih u prošlom broju Nova igra Nodes of Yesod Red Moon Fighter Pilot Staff of Karnath II I III Speed King Specijalna ponuda — najbolji COPY programi COPY 190 T T COPY — 190 blokova PIZZA TURBO — 200 bl PIZZA II — 202 bl ABC TURBO — 202 bl TURBO 2002 — 195 bl (autor M Zagar) + kaseta + poštarina = 1500 din Uputstva za početnike veliki popusti besplatan katalog Žuljavić Kemal Braća Kosorić 13 72220 Zavidovići tel 072/674-441

□ Za C-64 hitovi u kompletima po 15 + kaseta — 1500 din Imamo Elite Hacker Flight Simulator 2 School Daze Night Shade G—SOFT B Valjina 26/A 57000 Zadar tel 057/32-113

Prodajam programe za Commodore 64 Najjeftiniji paketi na zemlji 380 programa za samo 4900 din Rok isporuke dva dana na besplatnim compeci kasetama Tel 072/39-846

□ Commodore 64 Indiana Jones Hyper sports Pitfall III Rambo Petak 13 Forest of Doom Boulderdash III Aztec Tomb F 15 Hacker Basketbal III Garls of Fun Kaseta POKLON HARDVERSKI DODATAK. Komplet 2 Summer Games II Winter Games Beach Head II Kaseta POKLON SEME MIKROPROCESORA Komplet košta 1000 din oba 1800 din Sakač Rajko Novi Šor 86 21466 Kucura tel broj druga 021/725-060 Željko

□ PRODAJEM OKO 1400 NAJ PROGRAMA ZA KOMODOR 64 — NA KASETAMA CENA 22 DINARA KOMPLET 30 000 DINARA EKSPRES USLUGA BESPLATAN KATALOG N KALABA Slavke Đurđević B5-I/29 35000 Svetozarevo tel 035/27-118

o COMMODORE 64 — Veliki izbor kasetnih programa (20 najboljih u broju 11) Novi su Basketball 3 Boulderdash 3 Winter Games Walkiria 17 Witch a Couldron Supermen Commando Lords of Midnight Cylu Kliff Hanger Paranoid (ABC) Bounty Bob 201 Besplatan katalog savati za početnike P S Tonske kopija (preko 2 kazetofona) na držim Žuljević Kemal Braće Kosorić 13 72220 Zavidovići tel 072/874-441

o KOMDDOROVCII TRAŽITE LI Friday 13 Indiane Jones FGHT Hyper Sports Commendo Superman Staff 1—3 Impossible 2 Winter Games ? Javite se Locomotion Softwareu! Tražite katalog i uvjerite se! Parić Emil Rastočine 33 51000 Rijeka tel 051/511-032

• C-64 komplet D T a SUPER-TEST ZORO BOULDERDASH 3 DYNAMITE DAN GHOSTBUSTERS 2 PITSTOP 3 RAMBO FRIDAY 13 NODES DF YESOD Programi + kaseta = 100 dinara pouzećem Isporuka odmah Armen Ibrahimagić Cetinjska 12 71000 Sarajevo tel 071/456-329

o COMMODORE SUPER SOFT — hitovi na kazeti (RAMBO 2 CDMMANDD KUNG FU MASTER SKDDL DAZE KOKO BOULDER DASH 3 J Domjenović Albinijeva 2 41020 Zagreb, tel 041/687-081

COMMODORE 64
Kompleti sa kasetom 1700 dinara
Komplet 1
Boulder Dash III
America Road Race
Koko
Baseball III
My Chase II 2D
Wargames II
Jungle Deal
Nick Faldo Golf
Transformers
Ghostchaser
Trolls
Pitstop III (III zamena)
Komplet 2 Kung Fu Master Commando Willow Pattern Skool Daze, Superman Zorro Alien 8 Koko Rambo II Who Dares Wins II Wizards Lair Cylu
Čurčić Dušan Osmana Đikića 16a 11000 Beograd tel 011/762-022

o Izuzetna prilika — komplet od pet najboljih CDPY programa (PIZZA 202 TT — CDPY TURBO 200 ABC TURBO TURBO 199) + uputstva = 1000 din Domjenović Albinijeva 2 41020 Zagreb tel 041/687-081

JOKER SDFTWARE vam predstavlja samo najnovija i najbolja programe za vaš C-64 Poručite besplatan katalog na telefon 021/398-245 Jocić Dragan Put Bačkopalanačkog odreda 20/2 21000 Novi Sad

o KOMODORCII Najnoviji programi na kaseti uz najniže cene Tražite besplatan katalog Stanić Branko II bulevar 128 II ulaz 11070 Novi Beograd tel 130-684 III 152-083

o KOMODOROVCI već tri godina na tržištu dokazujemo svoju profesionalnost stručnost i solidnost Veliki izbor programa svrstanih u komplete što sadrže 20 do 60 programa a stoje 1000 do 1500 dinara plus kaseta Upute za početnike Isporuka odmah Tražite katalog (sa napomenom za CBM 64 i 50 d) skratite čekanje narudžbom preko telefona 041/417-052 FUTURE ORION 41000 ZAGREB Rubetićeva 7 Za veće narudžbe veliki popust!!

o Komodor 64 RAMBO 2 SKOOL DAZE SUMMER GAMES 2 COMMANDO KOKDTDNI WILF BDULDER DASH 3 i još mnogo novih programa po niskim cenama Adrović Miroslav Milutina Uskokovića 43 24000 Subotica tel 024/29-760

o Najnoviji svetskihit-programizaCOMMODORE 64 u kompletu
— FIGHTING WARRIDR (Ex Fist II)
— RARATERA (3D Karate)
— BOULDER DASH III
— CHUCKIE EGG II
— PITSTOP III
— FRIDAY 13th (Petak 13 )
— MASTER KUNG FU
— ZORRD
— BLADE RUNNER (popularni film)
— SUPERMAN
Komplet navedenih programa sa kasetom staje 1200 din Rok isporuka 48 časova Plaćanje pouzećem Čajkovski Karlo Anka Matić 3/3 11210 Beograd tel 011/711-358

o Za Komodor 64 kasetofon prodajem Cena po dogovoru Bata tel 021/21-758

CDMMDDORE 64 — Paket A RAMBO II KUNG FU MASTER COMMANDO SKDDL DAZE PYJAMARAMA III FRENKIE INDIAHA JONES FRIDAY 13 Paket B WINTER GAMES FLIGHT SIMULATOR II GIRLS FOR FUN BEACH HEAD II SUPERMAN STAFF III Cijena paketa + kazete + poštarina 1000 din Alan Đurić Edvarda Kardelja 20 50000 Dubrovnik tel 050/22-807

o Komodor 64 — prodajem programa (WHD DARES WINS 2 NDDES DF YESOD SKDDL DAZE RAMBO 2 DOMMANDD WINTER GAMES HYPER SPORTS) pojedinačno ili u kompletima Bulatović Vuksan Đerdapska 10 11000 Beograd tel 011/406-836

Commodore VOLVO SDFT — prodajem i menjam programe snimena direktno iz računara na kasetofon na kome nije menjan azimut Svakom naručiocu tri programa besplatno Milojević Ivan 38218 Leposavić

C-64 — TESLA SDFT — 30 programa + kaseta + poštarina 1000 din Nikolić Nebojša Đura Đakovića 2/1 34000 Kragujevac tel 034/66-481

o Prodajem programe za C-64 Zovite za besplatan katalog ELITE SWS 2 i druge igre čekaju na vas kod nas Bakić Vladan Polimska P+7 84300 Ivangrad tel 084/81-277

o Komodor — razdelnik DVATASET Priključite dva datasete na C-64 Presnimava sve programe — sa ili bez zaštite Preklopnik za dva režima rada Uputstvo garantovan kvalitet Kesler Viktor Rumenačka 106/1 21000 Novi Sad tel 021/334-717

o C-64 ALL STAR SDFTWARE — Komplet najnovijih igara WINTER GAMES (8 igara) CDMMANDO RAMBO II SUPERMAN KUNG FU MASTER WIEV TO A KILL BOULDER DASH II BRIAN BLDODAXE EXPLODING FIST SKODL DAZE ALIEN 8 + kaseta + poštarina = 1200 din Stojnić Boris Bratstva jedinstva 10 75000 Tuzla tel 075/213-964

o KDMDDOR 64 — besplatan katalog sa 1200 kasetnih programa HYPER SPORTS RAMBD 2 KUNG FU FRIDAY 13 WORLD CUP 2 SQUASH GHOSTBUSTERS 2 HACKER I ZORRD Niske cene popust Milan Stamenović Vladimira Nazora 77 18000 Pirot

o Prevedene igre za C-64 Četvrti paket — hitovi NDDES DF YESOD SKODL DAZE SORCERY GHETO BLASTER CIPHOID 9 CHOPPER ELIDON Svi tekstovi na ekranu su na srpskohrvatskom jeziku Uz paket dostavljamo uputstvo za apsolutne početnike i katalog Cena 1500 din Predrag Cvetković 11000 Beograd Radmila Rajković 12/29 tel 011/768 741

## GALAKSIJA

o Povoljno prodajem integralna kola 8264 (8K RAM) interfejs 1 za GALAKSIJU kao i čipova 74 LS sva podnožja (profesionalna) Dimitrije Stošić M Pijade 22 25250 Odžaci tel 025/74-448

o HARDWARE za prepravku galaksije u galaksiju plus veliki izbor mikroprocesora logičkih i linearnih kola Posedujem preko 1000 šema iz najnovijih stranih časopisa Naznačite koja vas oblast interesuje Jelić Dragan Jasenička 37 11420 Smed Palanka

o GALAKTIČARI! Ukoliko želite da se oprobate u najboljoj igri za svoju mašinu kupite igru Inspektor SPIRIDON! 4 6 KB 100% mašinca Autostart To što ćete videti u ovoj arkadnoj avanturi niste verovali da je moguće na GALAKSIJI Program + kompletno uputstvo 500 din Kada budete videli program videćete da vredi mnogo više! Dragan Janković Svetozara Markovića 60 19000 Zaječar tel 019/23-661

o PRDAJEM DALAKSIJU 8/6K ispravljač original kutiju ostalo 30 000 din CASIO FX750P nov plus dodatak FA-20 sa printerom i priključcima ima konstantnu memoriju 70 000 din Štefić Vaso Celovška 87 61000 Ljubljana

o Prodajem računar galaksija 8/6K Subotić Radovan Omladinskih brigada 182/32 11070 N Beograd tel 151-277

## QL

o QL — Prodajem programe uputstva literaturu Sva na jednom mestu (najjeftinije) Velikovski Kire Nikola Trimpare br 4/8 91000 Skopje

## AMSTRAD

o AMSTRAD — prodajem programe Praćenje RAM-a Word Processor Assembler/Disassembler Hisoft Pascal Transformatar 500 din Rolland Ghostbusters 300 din COPY programi Proton I I II Speedmaster 500 din Katalog Tomaš Kastelic Cesta na Vrhovce 7 61111 Ljubljana tel 061/267-832

o AMSTRAD — PRDFESIONALNI PREVODI PRIRUČNIK ZA CPC-464 1300 d MAŠINSKI JEZIK ZA POČETNIKE 1300 d LDCDMDTIVA BASIC 1200 d DEVPAC 700 d MASTERFILE 700 d PASCAL 700 d TASWDRD 400 d TRTICA ALEKSA STEVAHA LUKDVIČA 8 11000 BEDGRAD TEL 011/563-348

o Prodajem Šnajder 464 zeleni monitor Tel 023/522-518

o Amstrad — dve kasete + poštarina + programi (HYPER SPORTS DEVIL S CROWN ARABIAN NIGHTS FIGHTING WARRIDR RAID DVER MDSDOW B J SUPERSTAR CHALLENGE 1 2 SLAPSHDT HARD HAT MACK 3D GRAND PRIX) = 5000 din Jovanović Dragan Dragoslava Jovanovića 12, 37000 Kruševac tel 037/30-568

o DOMPUTER BOX SDFTWARE vam nudi komplet DOPY programa sa uputstvima i sa kazetom za samo 2000 n d odnosno bez kazete 1500 d Komplet PRDTON I I II SUPERPROTON I I II SIREN COPY TAPE SPEEDMASTER V 2 0 I TAPE MECHANIC CDPY Silvije Lukić Dimitrovljeva 8 41000 Zagreb tel 041/510 296

## ATARI

o ATARI 520ST programe i literaturu razmjenjujem Dajem besplatne stručna savjete za programiranje i korišćenje svih tipova ATARI računala Zainteresiranim mikro-klubovima poklanjam stare strane stručna mikro-časopise Zvonimir MAKDVEC tel 062-714-118 (do 14 30h)

ATARI SOFT-CLUB Zrenjanin Prvi u YU Veliki izbor programa svih vrsta i literatura Pozivamo vas na razmenu i na saradnju Besplatan katalog Lacmanović Dejan Sindelićeva 31/A 23000 Zrenjanin tel 023/31-665

## BBC/ELEKTRON

× Kupujem programe i literaturu za ACORN ELECTRDN Ponude slati na telefon 011/328-957

## BBC/ELECTRON

*o Poklonici kompjutera!*

*Kompjuteri Acorn BBC*

*prvi put u Jugoslaviji uvoznik „Partizan" — Čačak tel 032/51-710, 41-366 teleks 13745 13814*

o SBC/Elektron — gotovi programi — Inspiracija za samostalna programa Izbor od 350 programa. Mogućnost izmena Usluge štampanja Pravodi uputstva. Besplatan katalog Milutin Marković Sinđelićeva AC/20 37000 Kruševac

## HARDVER

o GALAKSIJA — SPECTRUM NARDWARE
Prodajam komplet čipova za proširanja Spektruma sa 16K na 48K (ili 80K) (9800 din ) te čipove 2764 (2600 din ) 27128 (3000 din ) 4164 (1100 din ) 8118 (2200 din ) 4416 (4200 din ) Z80A — CPU (2200 din ) Z80A — SIO (2500 din ) 8502 (4500 din ) 6522 (4500 din ) 8255 (2200 din ) AY3-8910 (5500 din ) AY3-8912 (5500 din ) TEXTOOL 28 (5500 din ) te ostala TTL i CMOS kola za gradnju galaksije i druge gradnja
Servisiram brzo kvalitetno i računara Spektrum i C-64
N Ćetković J Laskovara 1 42000 Vraždin, tel 042/38-56

o BLASTI — ubrzaće vaše basic programa i do 40 puta Traka sa BLAST COMPILER-om i BLAST TOOLKIT-om i opširno uputstvo na 35 strana — svega 1500 din
Uz MACHINE CODE FOR BEGINNERS uživaćete i sami u efikasnosti i brzini mašinskog programiranja Traka i opširno uputstvo na 65 strana — svega 1500 din
Oba programa — 2500 din
VRCA MILAN Zarija Vujoševića 79 11070 Novi Beograd

o SPECTRUM SOFTWARE STUDIO — Izbor od prako 1000 programa — svaki program sa uputstvom — veliki izbor literatura — knjiga i originalnih programskih uputstava na engleskom i srpskohrvatskom jeziku Spisak programa je besplatan za katalog sa oglsom poslati 200 din Brza i kvalitetna usluga — proverite
PAJNIĆ MIRKO STRAHINJIĆA BANA 56 11000 BEOGRAD tel 011/188-190 posle 15h

o HURRICANE SOFTWARE — veliki izbor najboljih i najjeftinijih programa kvalitetan snimak isporuka u roku od 24h Besplatan katalog naručite na adresu Milosavljević Saša Titogradska 43 36000 Kraljevo tel 036/332-580

## LITERATURA

o AMSTRAD Profesionaln prevodi PRIRUČNIK (1100) LOCOMOTIV BASIC (1200) MAŠINSKO PROGRAMIRANJE (1300) ZAJEDNO (3400) Prevedana uputstva za usložna programe DEVPAC TASWORD PASCAL QUILL MASTERFILE HOME BUDGET Pojedinačno (700) KOMPLET (3500) Snimljeni program (200) AMSTRAD FUTURE Bate Jankovića 78 32000 Čačak telefon 032/30-34

o OL PROFESIONALNI PREVODI OL-ARCHIVE (1500) OL-TOOLKIT (1200) OL-PASCAL (1500) OL-FORTH (1500) U KOMPLETU (5000) ZARIĆ SLOBODAN Bate Jankovića 78 32000 Čačak tel 032/30-34

o KOMODGR Profesionalni prevodi PRIRUČNIK (850) KAKO DA PROGRAMIRATE C-64 (1000) PROGRAMERS REFERENCE GUIDE (1300) MAŠINSKO PROGRAMIRANJE ZA POČETNIKE NA C-64 — Prevod 1985 g (1300) GRAFIKA I ZVUK (500) MATEMATIKA (1000) DISK SISTEMI I ŠTAMPAČI (500) UPUTSTVO ZA 1541 (700) SIMONS BASIC (850) PRAKTIKALK (750) EASY SCRIPT (500) MAE (400) VIZAWRITE (600) PASCAL (500) Svih 18 knjiga (po povlašćeno) ceni (10 000) NOVO NA TRŽIŠTU Prevod najbolja knjige MAPPING THE C-64 (2500) Na oko 200 stranica su objašnjena sve memorijska lokacija Isporuka u martu KOMPJUTER BIBLIOTEKA Filipa Filipovića 41 32000 Čačak tel 032/31-20

o SPEKTRUM Profesionalni prevodi MAŠINAC ZA POČETNIKE (1000) DISASEMBLIRANI ROM (1200) NAPREDNI MAŠINAC (1300) i nova knjiga SET INSTRUKCIJA ZA Z80 (2500) SVE ČETIRI (5400) Prevodi uputstava DAVPAC MEGA BASIC BETA BASIC ARTIST MELBOURNE DRAW EDITOR ASEMBLER MONITOR DISASEMBLER QUILL TASWORD FIFT Pojedinačno (500) Komplet (3500) Svih 10 programa snimljenih na kasetu (2000) KOMPJUER BIBLIOTEKA FILIPA FIIPOVIĆA 41 32000 Čačak tel 032/31-20

## RAZNO

Kupujem prvih pet brojeva časopisa Moj Mikro Tel 603-554

# „PISTE U NOĆI"

## KNJIGA KOJU PILOT NAMENJUJE PILOTIMA



Najzad je izašla iz štampe knjiga koju očekujete još od prošlog leta! To je novi „pilotski bukvar" Zorana Modlija, „Piste u noći". Knjiga je u međuvremenu promenila izdavača, što je bio jedan od značajnijih razloga ovog zakašnjenja. Autor vam se zahvaljuje na strpljenju a za uzvrat je dopunio knjigu nizom novih i uzbudljivih detalja.

„Piste u noći" anegdotski i lako razumljivo opisuju tehniku istrumentalnog letenja i vođenja aviona u savremenom vazdušnom saobraćaju, elektronsku opremu na zemlji i u pilotskoj kabini koja to omogućava, simulacija letenja na kućnim računarima — ali i udese nastale kao posledica nepridržavanja propisanog „bon-tona" i pilotiranja u hazardnim meteorološkim uslovima.

Zahvaljujući tome, moto knjige je KAKO OSTATI PILOT!

Stručni konsultanti i recenzanti knjige su saobraćajni piloti-kapetani JAT-a, kontrolori letenja i profesori Više vazduhoplovne škola.

Izuzetno zanimljivo i (ne samo za pilote) upotrebljivo štivo, rasuto je na 280 strana standardnog formata sa isto toliko ilustracija, u broširanom povezu i koricama u punoj boji.

„Bogato medijsko iskustvo autora, sklonost lakom i razumljivom kazivanju a uz sve to i njegovo profesionalno bavljenje letenjem, učinili su da dobijamo još jednu knjigu koja poput nekadašnja „Krilate katedra" sugestivno i nadahnuto mami za sobom novu armiju vazduhoplovnih zaljubljenika, pronoseći suštinsku ideju vazduhoplovstva da spaja obale okeana i pretvara ovaj naš globus u provinciju" (Recenzent Đorđe Jovanović, profesionalni pilot JAT)

NIRO „TEHNIČKA KNJIGA" 7 jula 28 11000 Beograd

Ovim neopozivo poručujem ________ primeraka knjige „Piste u noći" u izdanju „Tehničke knjiga" iz Beograda po ceni od 1600 dinara. Platiću prilikom prijema pošiljke — POUZEĆEM

Ime i prezime ____________________

Adresa ____________________

____________ ____________

Potpis Datum

# Testovi za „kreja“

Pod pojmom performanse računara većina ljudi podrazumeva broj operacija koje računar može da uradi za određeno vreme. Na žalost, za razliku od stalih performansi koje su nesumnjive (na primer količina ROM-a i RAM-a), podaci o brzini se znatno razlikuju od jednog prikaza do drugog. Razlog je u tome što racunar pri različitim zadacima pokazuje različite brzine, u zavisnosti od toga koji je kritičan resurs računara prouzrokovao izvestan zastoj u odnosu na maksimalne mogućnosti. Na taiko proizvođači računara na uporednim demonstracijama sa konkurentskim mašinama testiraju brzinu zadacima koji se izuzetno brzo izvršavaju na njihovom računaru, a na konkurentskom sporo.

Tabela 1.
Broj operacija po petlji

| PETLJA | broj operacija |
|---|---|
| 1 | 7217 |
| 2 | 8448 |
| 3 | 14015 |
| 4 | 9703 |
| 5 | 8347 |
| 6 | 9350 |
| 7 | 4573 |
| 8 | 4031 |
| 9 | 4910 |
| 10 | 4412 |
| 11 | 12002 |
| 12 | 11999 |
| 13 | 8846 |
| 14 | 9915 |

Stvarnu brzinu računara nije lako predstaviti kao jedan broj ili kao skup brojeva. Međutim, želja da se uporede različiti računari i da se ustanovi koji od njih je brži je toliko jaka da se veliki broj izuzetno kompetentnih institucija u svetu dao testove za merenje performansi sistema. Brzinski testovi su uglavnom na fortranu jer je brzina najbitnija kod numeričkih zadataka. Najcenjeniji brzinski test je test Livermorske laboratorije kojim možete da izmerite i brzinu vašeg računara. Ako nemate fortran kompajler, jednostavno translirajte programe na bejzik ili jezik čiji jezički procesor imate i pustite programe, izmerite vreme i izračunajte brzinu po trivijalnoj formuli:

Brzina (u flopovima) = broj operacija/vreme (u sekundama)

Broj operacija za pojedini zadatak (ukupno ima 14 zadataka i svaki u stvari predstavlja jednu petlju, što je učinjeno da bi računar pokazao svoju stacionarnu brzinu) dat je u tabeli 1, a petlje u tabeli 2.

Da biste mogli da se uporedite sa jačim mašinama, pogledajte tabelu 3. Ako vam se učini da vam računar nije dovoljno brz, pogledajte članak o superračunarima u prošlom broju „Računara“ i izračunajte odnos brzina/cena za vas i neki od superračunara i raspoloženje će vam se popraviti.

Ako ste i posle toga nezadovoljni, pišite nam i mi ćemo vas uputiti kako da nabavite neke druge testove, na kojima će se vaš računar pokazati možda boljim. Ako već nemate brži računar, imajte bar dobre rezultate na testovima.

**Nada Aleksić**

```
[illegible]
No 1        DO 1 k = 1,400
            [illegible]
No 2        [illegible]
No 3        [illegible]
No 4        [illegible]
No 5        [illegible]
No 6        [illegible]
No 7        [illegible]
No 8        [illegible]
          1 CONTINUE
No 9     DO 9 i = 1,100
         [illegible]
          9 CONTINUE
No 10       DO 10 k = 1,100
            [illegible]
```

```
            cr = br - px(8,i)
            px(8,i) = br
            ar = cr - px(7,i)
            px(7,i) = cr
            br = ar - px(8,i)
            px(8,i) = ar
            cr = br - px(9,i)
            px(9,i) = br
            ar = cr - px(10,i)
            px(10,i) = cr
            br = cr - px(11,i)
            px(11,i) = ar
            cr = br - px(12,i)
            px(12,i) = br
            px(14,i) = cr - px(13,i)
            px(13,i) = cr
      10 CONTINUE

No 11       x(1) = y(1)
            DO 11 k = 2,1000
      11      x(k) = x(k-1) - y(k)

No 12       DO 12 k = 1,999
      12      x(k) = y(k+1) - y(k)

No 13       DO 13 ip = 1,128
              i1 = p(1,ip)
              j1 = p(2,ip)
              p(3,ip) = p(3,ip) + b(i1,j1)
              p(4,ip) = p(4,ip) + c(i1,j1)
              p(1,ip) = p(1,ip) + P(3,ip)
              p(2,ip) = p(2,ip) + P(4,ip)
              i2 = p(1,ip)
              j2 = p(2,ip)
              p(1,ip) = p(1,ip) + y(i2+32)
              p(2,ip) = p(2,ip) + z(j2+32)
              i2 = i2 + e(i2+32)
              j2 = i2 + f(j2+32)
              h(i2,j2) = h(i2,j2) + 1.0
      13 CONTINUE

No 14       DO 14 k = 1,150
              ix = grd(k)
              xi = ix
              vx(k) = vx(k) + ex(ix) + (xx(k) - xi) * dex(ix)
              xx(k) = xx(k) + vx(k) + flx
              ir = xx(k)
              ri = ir
              rx1 = xx(k) - ri
              ir = ir - (ir/64) * 64
              xx(k) = ri + rx1
              rh(ir) = rh(ir) + 1.0 - rx1
              rh(ir+1) = rh(ir+1) + rx1
      14 CONTINUE
```

Znak & označava nastavak linije i piše se u koloni 5

Tabela 3. Performanse superračunara testiranih petljama Livermor laboratorije (u megaflopovima)

| petlja br. | HITACHI S810/20 | FUJITSU VP100 | FUJITSU VP200 | CRAY 1 CFT COMPILER | CRAY 1 CIVIC COMPILER | CRAY X-MP (UNIPROCESSOR) CFT 110 | CRAY X-MP (UNIPROCESSOR) CFT 113 + |
|---|---|---|---|---|---|---|---|
| 1 | 250 | 187 | 331 | 68 | 80 | 146 | 155 |
| 2 | 255-360 | 105 | 188 | 16 | 15 | 28 | 138 |
| 3 | 213-322 | 175 | 338 | 29 | 56 | 67 | 136 |
| 4 | 60-92 | 74 | 80 | 21 | 36 | 4 | 48 |
| 5 | 5-11 | 10 | 10 | 4 | 5 | 5 | 11 |
| 6 | 4-11 | 10 | 10 | 3 | 3 | 4 | 9 |
| 7 | 255 | 189 | 331 | 87 | 97 | 148 | 165 |
| 8 | 85 | 80 | 90 | 52 | 65 | 75 | 91 |
| 9 | 227 | 184 | 261 | 56 | 90 | 93 | 123 |
| 10 | 63 | 50 | 88 | 88 | 29 | 54 | 64 |
| 11 | 10 | 5 | 5 | 3 | 3 | 3 | 8 |
| 12 | 104 | 59 | 115 | 23 | 23 | 60 | 76 |
| 13 | 4 | 4 | 6 | 4 | 3 | 5 | 5 |
| 14 | 7 | 13 | 14 | 7 | 8 | 10 | 7 |